

Internet Rex

User Manual

Version 2.24

Features

Universal agent

Internet Rex supports a wide variety of internet transfer protocols. Rex is not only a mail agent, but also supports FTP, tossing files to a directory for use on FTP servers and BinkP for connecting to BinkD servers. If your BBS is connected to the net full time, Rex can also act as your server for FTP, mail, BinkP and more.

Universal compatibility

Internet Rex supports all the major email file transfer clients. It can read and write messages which will be understood by TransX, Fidonet2Internet mailer, Allfix, GIGO, Watergate and more. Whatever your downlink is using, Rex can talk to it.

Native operating system support

Internet Rex is distributed with native executables for the four major BBS operating systems: DOS, OS/2, Windows (including Win95, Win98, Win2000 and WinNT) and Linux. Native support means faster running programs and more seamless integration with the rest of your BBS system. In addition, Rex's configuration files are portable across operating systems: one set of files will work for all the versions of Rex available.

Frontend support

With 18 different frontend mailers recognised individually, Rex is sure to work with whatever system you're running. The most popular frontends in the list include FrontDoor (all versions) and compatible mailers, Binkley and compatible mailers, D'Bridge, Platinum Xpress, PCBoard, ViaMail, Intermail, SGMail and more.

Reliable, secure transmission

Internet Rex includes functions that will ensure that, despite the sometimes unreliable internet, your mail will always arrive at its destination. Built-in encryption can also be used to make sure that no one else can access the mail you're sending except the intended recipient.

Feature rich

A flexible, built in file request processor, a powerful FTP scripting language, netmail to email gating, firewall and network aware setup, automatic bundling and extraction of files: over and above the usual features you'd expect in an internet transport program, Rex adds extra functionality that lets you do more with your BBS and internet connection.

Table of contents

TABLE OF CONTENTS	1
INTRODUCTION	5
CONVENTIONS	5
WILDCARDS.....	6
OPERATING SYSTEMS.....	6
SYNTAX.....	6
SUPPORT.....	8
FIDONET BACKBONE CONFERENCE.....	8
EMAIL.....	8
NETMAIL	8
DOWNLOADING	9
THE WORLD WIDE WEB (WWW).....	9
FILE TRANSFER PROTOCOL (FTP).....	9
DOWNLOAD FROM BBS	9
FILE REQUEST (FREQ).....	9
EMAIL FILE REQUEST	10
INSTALLATION.....	11
UPGRADING.....	12
INSTALLING AND UPGRADING UNDER LINUX USING .RPMs	12
GETTING STARTED.....	14
THE ONLINE HELP SYSTEM.....	14
QUICK SETUP OUTLINE.....	15
GENERAL INFORMATION.....	16
REX'S BEHAVIOUR	17
<i>Operating system information</i>	17
Operating system dependent options	18
<i>Mail processing information</i>	19
<i>Interacting with TransX</i>	19
FILES, PATHS AND PROGRAMS.....	20
<i>Internet Rex files and paths</i>	21
<i>Frontend mailer setup</i>	22
ADDRESS MANAGER	26
<i>System addresses</i>	27
<i>Zone matching</i>	27
<i>Domains</i>	27
EMAIL AND FTP SETUP.....	27
<i>FTP setup</i>	27
<i>Email setup</i>	29

Table of contents

CONNECTING TO THE NET	34
<i>Connecting to the net under Linux</i>	35
<i>Connecting to the net under DOS</i>	35
<i>Connecting to the net under Windows 95/98/NT</i>	36
<i>Connecting to the net in OS/2</i>	37
SETTING UP A NODE	39
SETTING UP AN EMAIL LINK (TOSS TO/FROM MAIL)	40
<i>Sending mail</i>	40
<i>Receiving mail</i>	42
CONNECTING TO A BINKP SERVER (TOSS TO/FROM A BINKP SITE)	43
USING REX WITH YOUR OWN FTP SERVER (TOSS TO/FROM A DIRECTORY)	45
UPLOADING AND DOWNLOADING FROM AN FTP SERVER (TOSS TO/FROM AN FTP SITE).....	46
RUNNING REX	49
THE COMMAND LINE	49
COMMAND LINE SHORTCUTS.....	53
<i>Abbreviations</i>	53
<i>Merging</i>	53
INTERACTING WITH REX	54
DAEMON MODE.....	54
INTERACTING WITH REX IN DAEMON MODE	55
BBS ↔ INTERNET GATEWAYS	57
CONFIGURING THE GATEWAY	57
<i>How messages change going through the gateway</i>	58
<i>Message headers</i>	59
<i>Message flags</i>	60
SETTING UP NEWS SERVERS	61
SETTING UP GATES	61
<i>Basic gate options</i>	62
<i>Netmail ↔ email gates</i>	64
<i>Netmail ↔ domain gates</i>	65
<i>Echomail ↔ email gates</i>	65
<i>Echomail ↔ domain gates</i>	66
<i>Net gates</i>	67
<i>Mailing lists</i>	67
<i>Mailing list server</i>	68
<i>Newsgroups</i>	69
SOME SAMPLE GATES	69
<i>The personal email to netmail gateway</i>	70
<i>Gating a mailing list to and from an echomail area</i>	70
<i>Giving your users email addresses</i>	71
<i>Giving email addresses to a whole network</i>	72
GETTING TO KNOW REX	73

Table of contents

ROUTING	73
<i>route-to</i>	73
<i>hold</i>	74
<i>unhold</i>	74
<i>route.rex structure</i>	74
CUSTOMISING REX'S APPEARANCE.....	75
MULTIUSER EMAIL ADDRESSES.....	76
REMAPPING DRIVES AND DIRECTORIES	77
<i>Mapping drives to drives (DOS, OS/2, Windows)</i>	77
<i>Mapping drives to directories (Linux)</i>	78
FILE REQUESTS AND EMAIL FILE REQUESTS	78
<i>File request limits</i>	80
<i>Email file requests</i>	81
BETWEEN.BAT PROCESSING OF RECEIVED MAIL.....	82
AUTOMATIC FILE BUNDLING AND EXTRACTION.....	82
<i>Setting up the compression and decompression programs</i>	83
<i>Using file bundling</i>	84
DETECTION OF DUPLICATE MAIL	85
REX'S LOG.....	85
THE QUEUE EDITOR.....	86
RELIABLE EMAIL CONNECTIONS.....	88
SECURE AND ENCRYPTED MAIL.....	90
TRANSFER STATISTICS	91
ADVANCED NODE SETUP OPTIONS.....	92
REXFIX REMOTE MANAGEMENT.....	93
COMMAND LINE OPTIONS FOR THE CONFIGURATION PROGRAM.....	93
INTERNET REX'S SERVERS	95
REQUIREMENTS	95
THE FTP DAEMON.....	95
<i>Setting up virtual file systems (VFSs)</i>	96
<i>Adding a user</i>	98
<i>Additional FTP features</i>	99
<i>FTP daemon logging</i>	100
THE MAIL DAEMONS	101
<i>POP3 daemon setup</i>	101
<i>SMTP daemon setup</i>	103
Spool setup	105
Setting up mail users.....	105
<i>Setting up the finger daemon</i>	111
THE IDENT DAEMON.....	111
THE BINKP DAEMON	112
<i>Tips and tricks</i>	113
TROUBLESHOOTING.....	114
COMMON PROBLEMS.....	114

Table of contents

ERRORS FROM INTERNET REX.....	115
<i>Fatal errors (!)</i>	116
<i>Other errors (?)</i>	116
FREQUENTLY ASKED QUESTIONS.....	118
COPYRIGHT NOTICE.....	120
REGISTRATION	121
INDEX.....	122

Introduction

“In many of the more relaxed civilizations on the Outer Eastern Rim of the Galaxy, the Hitch Hiker's Guide has already supplanted the great Encyclopedia Galactica as the standard repository of all knowledge and wisdom, for though it has many omissions and contains much that is apocryphal, or at least wildly inaccurate, it scores over the older, more pedestrian work in two important respects. First, it is slightly cheaper; and secondly it has the words ***Don't Panic*** inscribed in large friendly letters on its cover.”

- Douglas Adams, *The Hitch Hiker's Guide to the Galaxy*

Welcome to the Internet Rex user manual. You have downloaded one of the most powerful BBS to Internet transport programs available, packed with features and useful tools. This manual will guide you through the setup, installation and use of Internet Rex with your frontend mailer and internet connection. Because of the wide range of features Rex includes, not all details on all of the possible configurations will be presented here; instead this manual tries to ensure that your installation is quick and easy and that you can get Rex up and running without problems. Some of the more advanced features are left to discussion in the Internet Rex technical manual, included with this distribution. All the information from both manuals is available in Internet Rex's online help.

If this is the first time you've taken a look at Internet Rex, you should go through the installation and setup guides to get you started. **Don't panic.** You don't need to fill in every field in every menu to use Rex. Decide what you want Rex to do, and use this manual to configure Rex to do it.

Users already familiar with Rex can find information about some of the more advanced features of Rex in the [Getting to know Rex](#) section of the manual.

Conventions

In order to make things easy to understand and let you find information quickly, a number of conventions are used throughout this manual. Text you would type into the computer will be presented in a typewriter style font. Text you should be typing yourself will be presented in **bold** typewriter font. For example:

```
C:\REX> rexcfg
```

 Important information will be signalled by a little pointer.

Field names (i.e. the highlighted parts of the menus that select what you'll be entering) will be represented in the text in *italics*. Field values (text you enter into the fields, or that you set the fields to) will be indicated by “quotes” around the values. Note that the “” signs don't actually form part of the value to be entered.

Introduction

Wildcards

Rex uses a system of wildcards similar to what is used in most operating systems. In places where you're allowed to give wildcard information, Rex will let an asterisk (*) match against any string of characters, no matter how long. A question mark (?) can also be used to match against any one character. For example:

<code>rex*</code>	Matches: <code>rex</code> , <code>rexw</code> , <code>rexp</code> , <code>rexls</code> , <code>rex.exe</code> , <code>rex.ovr</code> Doesn't match: <code>re</code> , <code>relx</code> , <code>arex</code>
<code>169:4*</code>	Matches: <code>169:4100/101</code> , <code>169:400/101</code> , <code>169:4</code> Doesn't match: <code>169:5000/101</code> , <code>169:1000/1</code> , <code>169:2200/4.0</code>
<code>rex?</code>	Matches: <code>rexw</code> , <code>rexp</code> , <code>rexl</code> Doesn't match: <code>rex</code> , <code>rex.exe</code> , <code>rexls</code>
<code>169:4??/*</code>	Matches: <code>169:420/101</code> , <code>169:400/1.0</code> , <code>169:444/1.1</code> Doesn't match: <code>169:4200/101</code> , <code>169:40/1</code> , <code>169:420.0</code>

Rex also processes [] style wildcards. Include a character set between the []s and Rex will match any one character from that set.

<code>169:4[1-790]00/*</code>	Matches: <code>169:4100/101</code> , <code>169:4500/101</code> Doesn't match: <code>169:4800/101</code>
-------------------------------	--

Operating systems

Internet Rex is available for four different operating systems. When there is information to be given which is specific to one of those operating systems, you'll see a notice of it to the side of the information.

DOS	This information applies to the DOS version of Rex only.
OS/2	This information applies to the OS/2 version of Rex only.
Win	This information applies to the Windows version of Rex only.
Linux	This information applies to the Linux version of Rex only.

Syntax

When giving commands and arguments to fields in Rex, a standard command syntax is used. This will let you tell which parameters are optional, which are required, and what options can be used in combination with others.

< > Angle brackets are used to represent parameters that must be present for a given

Introduction

- command.
- [] Square brackets are used to represent option parameters to a command. The options enclosed don't have to be present for a given command to work, but can be used to give additional information.
- | The straight line is used to present a list of valid choices for a given parameter. You can choose from the list of parameters presented.
- ... Ellipses are used to show that the previous choices (in either square or angle brackets) can be repeated as often as desired.

Brackets can be nested, so that their influence is exerted over whatever they contain.

Here are a few examples:

```
rex <-f filename>
Valid choices would be:
rex -f file.txt
rex -f another.file
```

```
Invalid choices would be:
rex
    The <-f filename> is required.
rex -f file.txt file2.txt
    Only one file can be specified after the -f.
```

```
rex [-f filename]
Valid choices would be:
rex
rex -f file.txt
```

```
Invalid choices would be:
rex -f file.txt file2.txt
    There can still only be one file specified after the -f.
```

```
rex [-f filename | -s filename]
Valid choices:
rex
rex -f file.txt
rex -s file2.txt
```

```
rex [-f filename [filename]...]
Valid choices:
rex
rex -f file.txt
rex -f file.txt file2.txt
rex -f file.txt file2.txt file3.txt file4.txt
```

Support

“’Tis not enough to help the feeble up,
But to support him after.”
- *Shakespeare, Timon of Athens*

We would like to make your use of Internet Rex as enjoyable and trouble-free as possible. To that end, if you’re having problems with Internet Rex, please contact us through any of the following support methods.

Fidonet backbone conference

Internet Rex has a support echo on the Fidonet backbone. Here you can get help from the author and from other users of Internet Rex for whatever problem you might be having. The echo to subscribe to is **IREX**.

Email

You can reach the author for technical support by sending mail over the internet. Send messages to:

cruden@cs.ualberta.ca

Netmail

Internet Rex’s main support BBS, Xanadu BBS, is reachable through a number of different nets:

Fidonet	1:342/806 or 1:342/820
BattleNet	169:4100/101
Sysop’s TechNet	111:1200/11

Downloading

“Something hidden. Go and find it. Go and look behind the Ranges –
Something lost behind the Ranges. Lost and waiting for you. Go.”
- *Rudyard Kipling, The Explorer*

Internet Rex is distributed in native versions for DOS, Windows, OS/2 and Linux. You can tell the operating system and version of an Internet Rex distribution archive by its name. Match the archive name to the following:

irexV###.zip

The ### is the version of Internet Rex. For instance, 100 would be version 1.00. The V is the operating system identifier. D is the DOS 16-bit version, E is the DOS 32-bit version, W is the Windows version, P is the OS/2 version and L is the Linux version. So the archive `irexd100.zip` would be Internet Rex version 1.00 for DOS 16-bit.

Internet Rex is available from a number of different sources. Choose the one easiest for you.

The world wide web (WWW)

Internet Rex's official homepage, with links for downloading is
<http://plaza.v-wave.com/InternetRex>

File Transfer Protocol (FTP)

Internet Rex's official FTP server is `xanadu.v-wave.com`. Internet Rex is available for download from the `/irex` directory. Old versions of Internet Rex are kept in the `/irex/oldversions` directory.

Download from BBS

The most recent version of Internet Rex will always be available for download online at the Internet Rex support BBS. That BBS is currently:

Xanadu BBS

Phone #s: 1-780-439-8364 or 1-780-433-3560

Accepting 2400 to 33600 baud callers 22 hours a day (from 4am to 2am, MST).

Telnet to: `xanadu.v-wave.com`

File request (FREQ)

Internet Rex can be file requested (FREQ'ed) from the Rex's distribution BBS. Currently that board is:

Downloading

Xanadu BBS - Fidonet address 1:342/806 or 1:342/820

The magic names to use when requesting Internet Rex are:

rexd	For the DOS 16-bit version.
rexe	For the DOS 32-bit version.
rexw	For the Windows (Win95, Win98, Win2K, WinNT) version.
rexp	For the OS/2 version.
rexl	For the Linux version.

Email file request

You can send an email file request to the Internet Rex email **FREQ** server and have Internet Rex sent back to you in a file attached to an email message. To email **FREQ** Internet Rex, send mail to:

xanadu@v-wave.com

The subject should be **FREQ** and the body of the message should contain the line **FREQ <version>**. Replace **<version>** with the keyword representing the operating system you would like. The keywords are:

rexd	For the DOS 16-bit version.
rexe	For the DOS 32-bit version.
rexw	For the Windows (Win95, Win98, Win2K, WinNT) version.
rexp	For the OS/2 version.
rexl	For the Linux version.

Installation

“A large number of installed systems work by fiat. That is, they work by being declared to work.”

- *Anatol Fiat*

If you're installing Internet Rex under Linux using the .rpm file, skip ahead to the installing under Linux section.

Installing Internet Rex is simply a matter of extracting it from the archive it was distributed in. You should have received Internet Rex in a PKZip archive. To install Internet Rex, create a directory where you would like to keep Rex. A good place would be the IREX directory:

```
C:> mkdir IREX
C:> cd IREX
```

Then change to the directory you created and unzip the archive Internet Rex came in. For instance, if you downloaded Internet Rex 2.24 for DOS, the archive name would be IREXD224.ZIP. So you might type:

```
C:\IREX> pkunzip c:\irexd100.zip
```

pkunzip is the program to run to extract files from a PKZip archive: the name of the program will likely be different on different operating systems. Use whichever program is appropriate for your operating system.

If you are using PKWare's PKZip for your particular operating system, you should see the authenticity verification information after unzipping the file. This would look like this:

```
...
Testing: REXCFG.EXE      OK  -AV
Testing: UPDATE.COM     OK  -AV
Testing: UPDATE.DOC     OK  -AV

Authentic Files Verified!   # TLH613
Charles Cruden
Khan Software
This is Internet Rex 2.24.
Get the latest version of Internet Rex from
  http://plaza.v-wave.com/InternetRex.

C:\IREX>
```

If you are using PKWare's PKZip and you don't see this authenticity information, be aware that the archive you've downloaded is not a copy of the original distribution file, and may have been modified.

Installation

Linux The .ZIPs Rex is distributed in do not preserve file case properly under Linux, nor do they preserve file permissions. You should extra Rex from .ZIP archives using the command `unzip -L <filename>` to ensure that the files created have all lower case filenames. You should also be sure to `chmod u+x rexl rexls rexcfgl update` to ensure that these executables are actually executable.

Upgrading

If you already had a copy of Internet Rex installed, you'll have to do one more thing to ensure that the information you've already entered into Rex is also entered into the new version. After unzipping the archive into your Rex directory as described above, change to that directory and run the update program. That is:

```
C:> cd IREX
C:\IREX> pkunzip c:\irexd224.zip
C:\IREX> .\update.com
```

`update.com` is a DOS program, so you should run this in a DOS window if you aren't running DOS itself. `update.com` may give you further instructions to upgrade your version of Rex.

Linux The update program for Linux is simply called `update`. You should run `./update` to update Rex, as many distributions of Linux include a system utility called `update` which will usually appear first in your path.

Installing and upgrading under Linux using .rpms

Internet Rex for Linux is available as both a .ZIP file and a .rpm file. The .rpm file is the preferred package format for Linux. To install and upgrade Internet Rex under Linux, use the `rpm` utility with the `-U` command.

```
{machine}: rpm -U InternetRex-2.24.i386.rpm
```

This will update a previous installation or create a new one. By default, Rex and all its documentation will be installed to the directory `/usr/local/InternetRex`. If you would prefer to have Rex in a different directory, use the `--prefix` switch to specify the directory you want to install or upgrade Rex to. For example, to upgrade the Rex installation in `/home/Rex`, the command would be:

```
{machine}: rpm -U --prefix /home/Rex InternetRex-2.24.i386.rpm
```

Installation

- ☞ Note that the `--prefix` switch is broken in versions of `rpm` up to 3.0.1. You will need at least that version for the `--prefix` switch to work. Use `rpm --version` to find out what version you have.

Getting started

Despite the wide range of features Internet Rex offers, you can get up and running with only very little information setup. This section will help you configure Rex so you can start uploading and downloading mail as quickly as possible. More advanced features will be presented in the [Getting to know Rex](#) section of this manual.

Once you've installed Internet Rex, you need to run the configuration program to let Rex know a bit about your system. Change to your Rex directory and run Rexcfg.

```
C:> cd IREX
C:\IREX> rexcfg
```

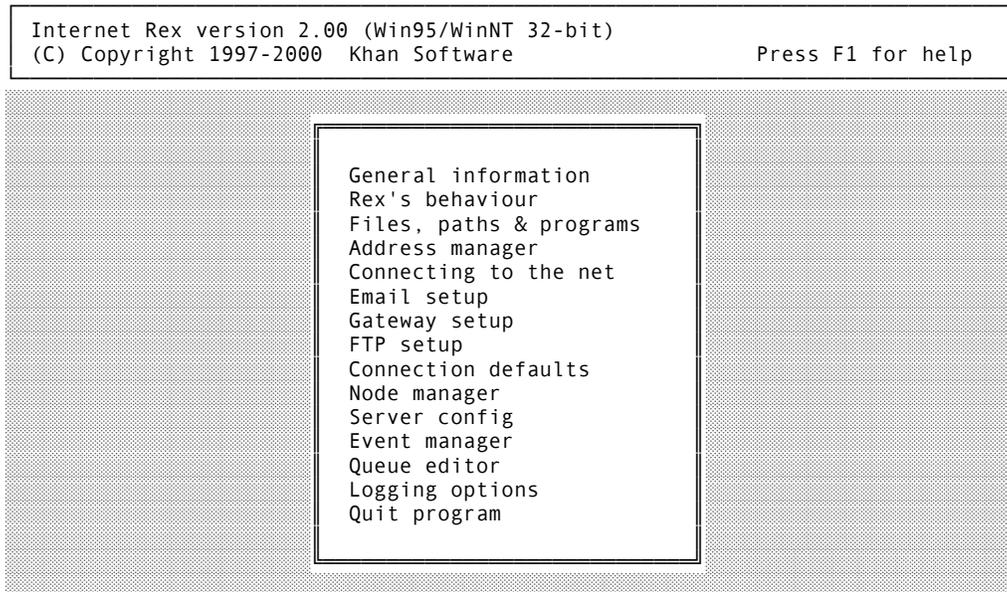
The configuration program is named differently in each operating system, so that Rex can be installed for many operating systems in the same directory.

OS/2 The configuration program for OS/2 is called `rexcfgp`.

Win The configuration program for Windows is called `rexcfgw`.

Linux The configuration program for Linux is called `rexcfgl`.

With the configuration program up and started, you should be presented with Rexcfg's main screen:



The Online Help System

Before getting into configuring parts of Rex, it's best to first get acquainted with Internet Rex's online help system. This system provides context sensitive help from anywhere in the configuration program to help you understand what each part of Rex does. You can access it at any time by hitting the **F1** function key.

Getting started

If you enter the online help system at the main screen, you'll be presented with Rex's general help menu. As you read through the text there, you'll notice certain words are highlighted in different colours. One set of words is in blue on white, and others are in light blue. These are **help links**, a little like links in a web page. You can highlight different links using the left and right arrow keys, or by clicking on them with your mouse.

Linux Mouse support is not available in the Linux version of Rex.

If you click on the blue on white choice or hit **enter** while it's highlighted, you'll follow the link to a different help page, one that has information on the keywords in the link. For instance, if you let the help link sit on the words **enter** or **f1** at the top of the screen and hit the **enter** key, you'll go to the Help on help screen, explaining how to use the online help system. To go back to the main help screen, hit the **B** key to backup.

Larger help screens can be scrolled up and down using the up and down arrow keys. You can advance a whole page up or down using the page up and page down keys, or by clicking on the scroll bar at the right hand side of the screen.

Keys in the online help system

F1	Accesses the online help system from anywhere in the configuration program.
←, →	Change the current choice of help links.
↑, ↓	Scroll the current help page up or down one line.
PgUp, PgDn	Scroll the current help page up or down one screen.
Enter	Follows the current help link to its help screen.
B	Backs out of the current help screen.
I	Brings up an index of help topics.

Quick setup outline

To get Rex working for you as quickly as possible, there are just a few steps to go through to get everything configured:

1. Fill in general information about you and your BBS in the [General Information](#) screen.
2. Configure how you want Rex to deal with netmail and processor performance in [Rex's Behaviour](#).
3. Tell Rex about the paths you've installed it to and the paths your mailer uses in [Files, Paths and Programs](#).
4. Setup your system's netmail addresses in the Address manager.
5. Depending on what sort of transmission methods you'll be using, setup information about your email addresses in [Email setup](#) or about how you'd like FTP connections to work in [FTP setup](#).
6. If you're using a dialup connection to the Internet, or a LAN connection under DOS, configure it in the [Connecting to the net](#) section.

Getting started

Once those steps are taken, it's just a matter of adding in information about the nodes you'll be connecting to in the *Node manager*, or the gates you'll be using in the *Gateway setup* and you're set.

General Information

The first choice in the menu after running *Rexcfg* is *General Information*. Choose this option, and you'll be presented with the following screen.

```
Internet Rex version 2.00 (Win95/WinNT 32-bit)
(C) Copyright 1997-2000 Khan Software                               Press F1 for help

[■] General information
Sysop name                Joe Sysop
System name               Joe's BBS
System location           Somewhere out there
System flags              CM,IBN

Password password        *****
User level                Beginner

Registration string
Registration key

This copy of Internet Rex is unregistered (3 nodes)
Server functions are limited to 2 users.
```

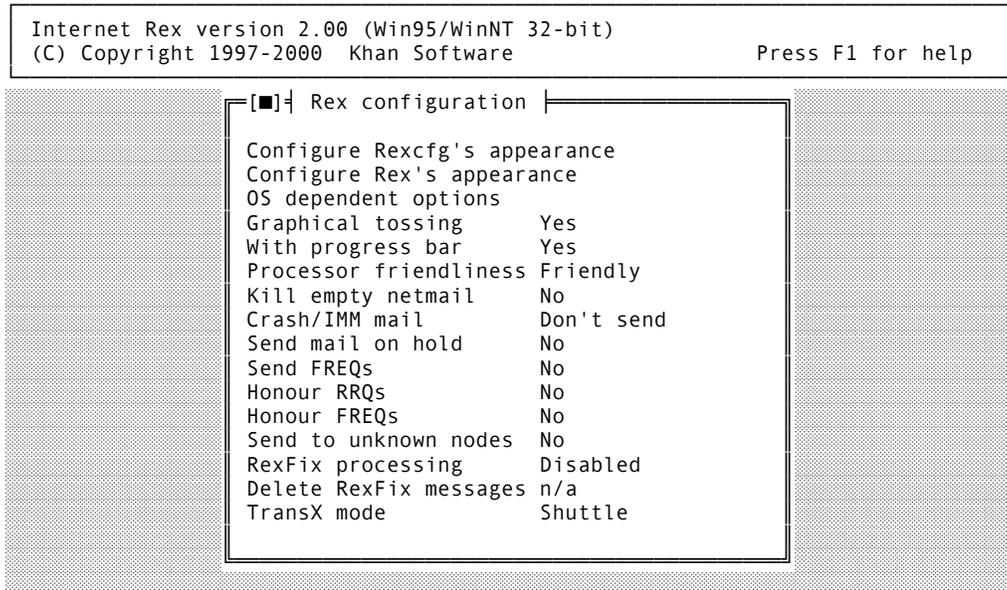
The General information menu lets you enter general information about yourself into Rex. The sysop name field here will be used as part of the From: field when sending email messages and will help link your registration information to your name, so no one else can use your registration key. The system name identifies your BBS to Rex, again for use in email messages and in establishing BinkP sessions. System location and flags give more information for BinkP sessions.

The *Password password* field isn't a typo: it's a password to protect the passwords you setup in Rex. If you're not the only one with access to your computer, you may want to prevent others from learning or modifying your passwords. If you enter a password here, *rexcfg* must be started with the argument *-ppassword:* if it isn't, the person running the program won't be able to read or modify any password anywhere in the program.

Because of the number of things that can be setup with Rex, there is a user level setting in this screen. Set this to your familiarity with Internet Rex by cycling through the different levels with the **enter** key. (Beginner is usually a good choice for someone just setting up Rex.) The more advanced the user level, the more options will be available for you to configure elsewhere in the program, the less Rex will try to configure itself. This provides you a bit more flexibility, but some of the options can cause unwanted results if not used carefully. When changing user levels, read the online help on new options to make sure you understand what they do.

Getting started

Rex's Behaviour



This menu lets you configure the more program unique properties of Rex with respect to things like playing with the operating system and reading your outbound mail queue.

Operating system information

The operating system related choices are *Graphical tossing*, *With progress bar*, *Processor friendliness* and *OS dependent options*. *Graphical tossing* and *With progress bar* also setup how Rex will appear when it's actually run. *Graphical tossing* makes Rex use a more informative graphical representation of its actions when it's reading/writing inbound/outbound mail. Turning it off will reduce processor usage, but make things a little less interesting to watch while Rex is running. *With progress bar* can be turned on when graphical tossing is used: it makes Rex display progress bars while uploading and downloading files so you know how much is left to do. Again, this incurs a bit of a processor hit. Finally, *Processor friendliness* can be used to control even further how much processing time Rex takes. The more processor friendly you make Rex, the longer Rex will take to run, but the more time other programs will get to run while Rex is running. If you're running Rex at the same time as your BBS, giving more time to other programs will mean your users will notice less lag while Rex is running.

DOS DOS isn't a multitasking operating system (you can only run one program at a time). Because of that, the *Processor friendliness* setting won't have any effect on the running time of Rex in pure DOS, so feel free to set it to whatever level you like. This isn't the case if you're running the DOS version of Rex under a multitasking operating system (like in a DOS window under Windows, or in a window of DesqView).

Getting started

Operating system dependent options

Some of Rex's configuration options only apply to particular operating systems. You should setup those options which apply to your OS.

DOS

The *Swap to EMS/disk* setting controls whether Rex will swap itself out to EMS (or disk, if there isn't enough EMS memory) when running other programs. For most DOS users, setting this to "yes" will cause no problems, and will mean that the programs Rex tries to run (such as compressors/decompressors) will have a great deal more conventional memory to run with. With some memory managers, the EMS/disk swapping causes problems: if you later find that there are crashes while trying to run other programs in Rex, turn this setting off.

Use EMS controls whether or not Rex itself will use EMS while running. Using EMS can drop Rex's conventional memory usage up to 80K, but may cause problems in some rare cases. It's enabled by default.

OS/2 and Windows

Rex can take advantage of these operating system's multitasking abilities by sending and fetching mail for many nodes at once. You can control how many tasks Rex will try to perform at once with the *Maximum inbound threads* and *Maximum outbound threads* fields: each of these can be set as high as 64 or as low as 1. You can also control how many of each type of task (tossing to email, a directory, an FTP site or a BinkP site) Rex will try to perform. This can be useful if you have a slow hard drive (in which case you don't want too many toss to/from directory tasks going at once) but a fast internet connection (so more FTP/email/BinkP tasks can be handled). Each of these fields can be set to a value between 1 and 16.

Linux

Like OS/2 and Windows, Rex can use multitasking in Linux to toss mail for multiple nodes at once (see above). Unlike OS/2 and Windows, Linux has some file system features that are rather important, namely permissions and case sensitive filenames.

Rex has two fields dealing with case sensitive filenames: *Inbound file case* and *Aggressive file find*. *Inbound file case* determines what case will be used to save inbound files: this can come into play if, for example, you receive mail bundles in uppercase names but your mail tosser expects lowercase ones. In that case you'd set the inbound file case to "Lower case". You can force upper case as well, or have Rex leave the files in the same case they were received in. *Aggressive file find* is useful for when you're running DOS programs under Linux which create filenames in one case and pointers to them in another. For example, some Binkley mail tossers create bundles with lowercase filenames, but put the names in as uppercase in the `.?l0` files. Enabling aggressive file finds tells Rex to look for files in both the expected and lowercase versions. While this will find more files, it may also slow rescanning down a bit.

Permission related fields are the *umask* and *User ID* for Rex. The *umask* sets Rex's umask for both the configuration program and Rex itself. This will be bitwise anded with

Getting started

your umask to determine the final set of permissions on the directories and files Rex creates. The user ID field is mostly for when you want to run some of Rex's daemons. The well known services (FTP, email, finger, etc.) under Linux require that the process running those servers be user ID root, but you probably don't want Rex to interface with your BBS as root. To get around this, if Rex finds a user ID entered here, it will try to use that user ID for BBS functions when it is run. This way you can run Rex as root but have it interface with your BBS with another user ID.

Mail processing information

The mail processing settings are *Kill empty netmail*, *Send crash/IMM mail*, *Send mail on hold*, *Send FREQs*, *Honour RRQs* and *Honour FREQs*. Because Rex doesn't act quite like a normal mailer, you can determine how Rex will act when it encounters certain types of messages.

Kill empty netmail applies to Rex's behaviour while reading outbound netmail folders. If you enable this, Rex will simply delete outbound messages to nodes in the Node manager if they are empty. A lot of messages fall into this category; they are usually created by mail tossers or inter-BBS door games, and are simply there to have the mailer send a file - the actual message doesn't need to be sent.

Send crash/IMM mail, *Send mail on hold* and *Send FREQs* tell Rex how to deal with crash or immediate mail, mail on hold and file requests. You may want to have your mailer send crash or immediate mail, or keep mail on hold for someone to poll and pick up. Sending the mail through Rex could confuse the situation. *Send crash/IMM mail* also has a third setting (besides "Send" and "Don't send"): "Send immediately". If you run Rex in daemon mode, "Send immediately" means Rex will try to send crash mail as soon as it sees it rather than waiting for the next mail event. *Send FREQs* lets you choose whether or not you want Rex to send file request messages: some of the programs Rex can talk to can't handle or don't properly handle incoming file request messages, in which case it would be better to send these with your mailer.

Honour RRQs and *Honour FREQs* apply to inbound messages. RRQs are "return receipt request" messages, messages which ask the receiver to send a receipt confirming they've arrived properly. These are poorly supported in most mailers and not generally used: it's better to use Rex's native system of receipts to transfer messages than this method, but it is supported if absolutely necessary. *Honour FREQs* determines whether or not Rex will process inbound file request messages it receives. If you have an external file request processor, you should turn this off so that it can process the messages itself. Otherwise, Rex will handle the messages using its own internal file request processor. (Note that this doesn't apply to email FREQ messages, which only Rex itself can process.)

Interacting with TransX

This menu includes one other option, *TransX mode*. This setting determines how Rex will deal with inbound/outbound TransX email. If you don't intend to transfer mail with a node using TransX, don't worry about this setting.

Internet Rex can work with TransX messages in two different ways: it can either act as a shuttle for them, uploading and downloading any messages it finds for or created by TransX

Getting started

(this is “shuttle” mode); or it can process the TransX messages itself (this is “compatible” mode).

If Rex is running in shuttle mode, you’ll need a copy of TransX to process and create TransX messages. In this case, Rex replaces TXMailer in the setup for TransX. You’ll have to tell Rex a bit about your TransX setup for this setting to work: this extra information should be entered into the *Files, paths and programs* → *TransX setup* menu after you’ve configured TransX.

If you decide to run Rex in compatible mode, it can read and write messages similar to those created by TransX 1.5 (and compatible with TransX up to 3.0 at this time). “TransX compatible” will be a transport type available for nodes whose outbound mail is to go out over email. This lets you talk to sites using TransX without having to own/register TransX yourself.

Files, Paths and Programs

This menu is where you configure how Rex interacts with various other programs and where Rex should be looking for files and mail on your hard drive. There are two parts which need to be configured to get Rex up and running: *Internet Rex files and paths* and *Frontend mailer setup*. Remaining options will be explained in the [Getting to know Rex](#) section of this manual.

Linux Linux users who are using Rex in combination with programs running in a DOSemu window (i.e. who are creating stuff with DOS style drives) should also setup drive to directory maps in the *Drive mappings* → *Map drives to directories (Linux)* (see [Remapping drives and directories](#)). This will let the Linux version of Rex figure out where to find files referenced with a DOS filename.

Getting started

Internet Rex files and paths

```
Internet Rex version 2.00 (Win95/WinNT 32-bit)
(C) Copyright 1997-2000 Khan Software                               Press F1 for help

[■] Internet Rex files and paths
Log filename                c:\irex\rex.log
Temp. files directory      c:\irex\temp\
Queue directory            c:\irex\queue\
Holding directory          c:\irex\hold\
Semaphore directory        c:\irex\sems\

Inbound history file       c:\irex\inbound.his
Outbound history file      c:\irex\outbound.his
History file format        .HIS style
Lines of history to keep   500

Received mail semaphore    c:\fd\sema\fdxit6.50
TransX mail semaphore      c:\fd\sema\transx.now
Run BETWEEN.BAT/.CMD      No
Always run BETWEEN.BAT
```

Rex uses a number of different files and paths while it's running to keep track of mail sent, create temporary files and process inbound mail. They are all configured at this menu.

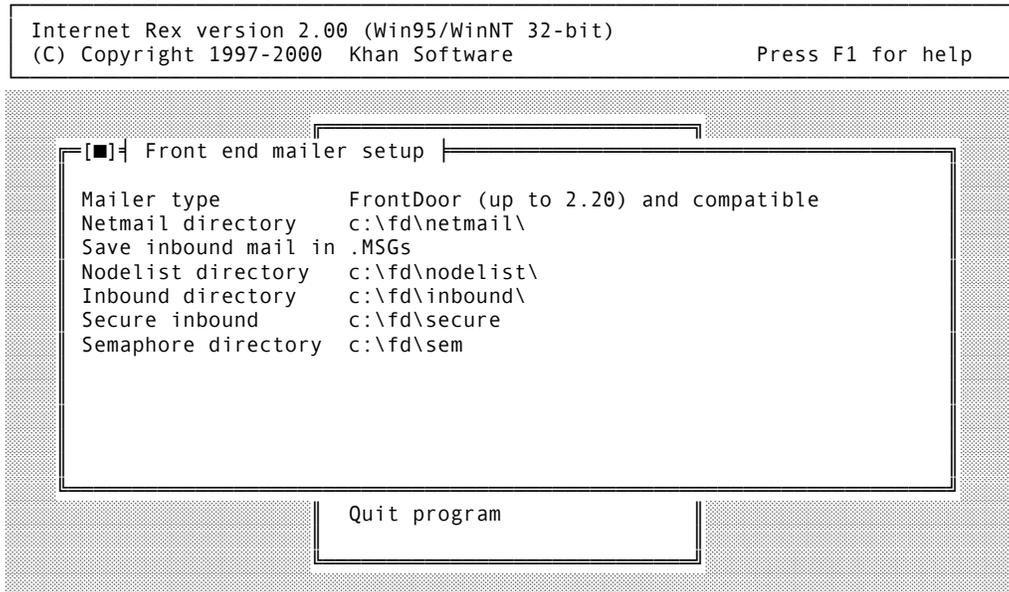
☞ All the files and paths you specify in this menu should be full file and path names. That is, they should begin with a drive letter (for DOS, OS/2 or Windows users) or / character (for Linux users). `C:\irex` and `c:\irex\irex.log` are full paths; `\irex`, `..`, and `.\irex` are not. If you'd like files to be accessible in the same place on many drives, you should take a look at the drive mapping feature of Rex (see [Remapping drives and directories](#) for more information).

The *Log filename* tells Rex where to store logging information while it's running. These logs will show information about what files have been transferred, possible configuration problems and so on. The *Temporary files directory* is where Rex will create any temporary files it might need to while running; if possible, this directory should be on a RAM disk to speed up processing. Rex uses the *Queue directory* to store files that haven't been sent yet and to hold on to files that have been sent in case people request resends later on. It also stores information about what files have been received here. The total size of files in here can get quite large if you're sending mail to a number of different nodes. Finally, the *Holding directory* is where Rex creates larger temporary files and where it stores partially completed FTP downloads. You must specify each of these directories or files in order for Rex to work properly.

The remaining options are covered later on as they aren't usually needed immediately.

Getting started

Frontend mailer setup



Here is where you configure Rex's interaction with your frontend mailer system. Rex supports a number of different mailers natively, so the appearance of the menu choices here may vary quite a bit, depending on what mailer you choose in *Mailer type*. Choose the mailer type that most closely matches your system; the most common mailers are listed already. Users of Xenia, Argus and McMail should choose Binkley compatible as their mailer type. If your mailer isn't listed by name and doesn't fall into one of the "compatible" categories, try the *.PKT mailer format. Despite the different ways many mailers work, the first few fields to setup are the same for all of them.

Many mailers and most interBBS door games require the presence of a netmail directory where Fido style *.MSG files are created. You should specify the full path to this directory on your system in *Netmail directory*. You can control what format Rex will use to save inbound messages for you in the *Save inbound mail in* field. On most BBSs it will be possible to import or directly read netmail saved to *.MSG files, and this is the default value for this field. If your board doesn't support this format directly, you may instead want to save the messages in a .PKT for your mail tosser to import into the BBS; four different types of .PKTs are supported - "Type 2+" is recommended for most setups as it's the most widely used.

The next field, *Nodelist directory*, is optional. If your mailer uses a FrontDoor, Intermail, V7+, BBBS, SGMail or GoldEd style compiled nodelist, Rex can use the nodelist to automatically configure nodes that have IP flags in their entries. Set this to the full path to your nodelist index. (For FrontDoor or Intermail style nodelists, this should be the path to your nodelist.fdx file; for V7+ nodelists, nodex.ndx; for SGMail nodelists, nodelist.sg; for BBBS nodelists, nodelist.idx; and for GoldEd nodelists, goldnode.gxn.)

For most of the mailers listed, the next two fields are *Inbound directory* and *Secure inbound*. These fields should point to the paths your mailer uses to store files from regular and secure connections. (The *Secure inbound* field is only required if it's different from the *Inbound directory*.) The field after that is usually *Semaphore directory*. This field is only required if you run a multinode system or if Rex will be running at the same time as your

Getting started

mailer. It should point to the directory where your mailer reads and writes semaphores: if your mailer doesn't have a specific directory for this, it's usually the directory where the mailer was installed. If you plan on running Rex in [Daemon mode](#) you should not only configure a semaphore directory, but also read the online help for more information on the semaphores Rex will be looking for, especially if your mailer isn't specifically listed among those Rex supports.

Configuring Binkley and compatible mailers and Portal of Power

Binkley systems and Portal of Power have an additional field to fill in, namely the mailer's *Outbound directory*. This should be the root of your mailer's default outbound directory: be careful not to add in zone information at the end of it. It should be something like `c:\binkley\outbound`, not `c:\binkley\outbound.001`.

If you're using a Binkley compatible mailer with multiple outbound domains (e.g. your mailer has not only the standard `c:\binkley\outbound` directory, but also something like `c:\binkley\mynet.033`), you should configure the additional domain directories in the Address manager later on. Don't add them here.

Configuring D'Bridge

The additional field for D'Bridge is the path to D'Bridge's *Outbound queue*. Rex will look in this directory for C, I, N, H and Q style queue files.

Configuring T-Mail

Internet Rex supports some of T-Mail's advanced features like filebox directories. If you are using these, you can enter the *Filebox directory* here and Rex will search it for outbound files.

T-Mail can run in either FrontDoor compatibility mode or Binkley compatibility mode. If you're using T-Mail in Binkley mode, you'll have to setup the *Binkley outbound* directory. Read the section on *Configuring Binkley and compatible mailers and Portal of Power* for more information on filling this field in.

Configuring AdeptX

AdeptX has an *Outbound directory*, something like Binkley's. Set this field to the full path to AdeptX's outbound directory.

Getting started

Configuring ViaMail

Setup of ViaMail proceeds a little differently than most mailers. Rex doesn't support a semaphore directory for ViaMail, as ViaMail should do all the mail processing itself - there are no concurrency issues with ViaMail.

In order to use ViaMail with Rex, you'll have to setup outbound mailbox directories for each of the nodes you want to send mail to. To do this, add each node into ViaMail's node manager first. Then, under the *Output format/mode* in ViaMail, select "Raw PKT/Alt path" or "Compressed PKT/Alt path" and choose a directory for "Alternate mail path". Note the "Alternate mail path" you choose for each node, and when configuring that node in Rex's node manager later on, put that directory in the *Mailbox directory* field for that node. That will tell Rex where to go to find outbound mail for that node.

It's important that you allow ViaMail to export the mail for each node and not have other programs do this for you. Otherwise, the messages for those nodes will accumulate in ViaMail's outbound queue and will have to be purged manually.

Configuring KBBS

KBBS's outbound is quite similar to Binkley's, except it has a *Zones directory* to configure instead of an outbound directory. You should set this to the directory where KBBS has the *zone** directories for its outbound. For instance, if mail for zone one were in `c:\kbbs\outbound\zone1`, you would set this to `c:\kbbs\outbound`.

Configuring PCBoard

Rex requires that you be using PCBoard version 15.22 or higher in order to work correctly. PCBoard stores parts of its outbound in an index file called `FIDOQUE.DAT`. You'll need to specify the path to this file in *FIDOQUE.DAT path*; usually this is just the directory where you installed PCBoard.

Configuring Platinum Xpress

Platinum Xpress's setup is quite different from most mailers. Since Platinum Xpress doesn't support a different inbound for secure and insecure connections, but does have different inbounds for files, mail and .TIC files, the *Inbound directory* and *Secure inbound* fields have been replaced with *Inbound files directory*, *Inbound mail directory*, and *Inbound .TIC directory*. Set these to the directories you have setup in Platinum Xpress to receive inbound files, mail and .TICs. (If your mail or .TIC directories are the same as your file directory, you can leave those fields blank.)

Instead of a semaphore directory, Rex communicates with Platinum Xpress by means of locks on its .DAT files. It also looks through Platinum Xpress's outbound mail and file queues for data destined for systems in the Rex's node manager. For that reason, it needs the path to PX's *.DAT files. These should be in PX's *System directory*, usually something like

Getting started

c:\px\system. This directory should contain the file PXOBMAIL.DAT and either PXOBFIL.DAT or PXATTACH.DAT, depending on whether you're using the DOS or Windows versions of Platinum Xpress.

Internet Rex also needs the directories where Platinum Xpress stores outbound .PKTs and arcmail bundles. These should be entered into the *Packet path* and *Arcmail path* respectively. (Often these two are the same, usually c:\px\outbound.)

Finally, Rex needs to know what version of Platinum Xpress it is working with, the DOS or the Windows version. Note that although Rex can talk to the DOS version of Platinum Xpress under any operating system, you must use the Windows version of Rex with the Windows version of Platinum Xpress.

Because Platinum Xpress's system of storing mail is closely linked with Wildcat, there are a few things that need to be done before running Rex to get mail correctly exported from Platinum Xpress. In order to get netmail out of the Wildcat message base and into Rex, you must use the PXNet program included with Platinum Xpress. Before running Rex, you must run `pxnet /scan`, and then `pxnet /pack <address>` for each address you want to export netmail to. After this, you must run `pxnet /sent <address>` to mark the mail as sent. So a sample batch file for running Platinum Xpress with Rex might look something like this:

```
cd c:\px
pxnet /scan
pxnet /pack 1:234/567
pxnet /pack 2:345/678
cd c:\irex
rex.exe
pxnet /sent 1:234/567
pxnet /sent 2:345/678
```

This would pack any outbound netmail to 1:234/567 and 2:345/678 into .PKTs which Rex would then pickup from the *Packet path*.

☞ Note that netmail exported by PXNet does not currently correctly export file attaches. This means Rex will not be able to find any files attached to netmail exported by PXNet. If you want to send a specific file to someone, you can do so by either creating a file attach netmail in the *.MSG netmail directory, which Rex can read correctly; by creating a mailbox directory for the node you want to send files to, putting the files in that directory then adding the mailbox directory to the node's entry in Rex's node manager; or by using `pxqedit` to add a file to Platinum Xpress's outbound file queue (choose *Edit file queue* while in `pxqedit` to do this).

Configuring FrontDoor 2.25+

FrontDoor versions higher than 2.25 support an advanced mail queue called the static queue. Internet Rex can read this queue and send out files from it accordingly. If you are using this feature of FrontDoor 2.25+, specify the path to the queue file (`filqueue.fd`) in the *FILQUEUE.FD path* field. Usually this is the same directory you installed FrontDoor to.

Getting started

Configuring QFront

Like PCBoard, QFront uses a queue file as part of its outbound structure. Rex needs the path to this file specified in *QUEUE.DAT path*. Rex also needs the directory where QFront builds outbound arcmail bundles (e.g. *.SU0, *.MO1, etc.) specified in the *Arcmail path*.

Configuring *.PKT mailers

If your mailer doesn't fall into any of the other mailer categories in Rex's *Mailer type* field, you may still be able to use Internet Rex with your mailer. To setup your mailer as a *.PKT mailer, it must be creating or be able to create outbound mail in the form of *.PKT bundles. Rex then requires the path to these bundles to be specified in the *Packet path*. If you find that Rex is giving error messages saying that it is unable to locate file attaches, you may also need to specify a *Default path*. This should be the path where the missing file attaches were located; usually this is the mailer's outbound directory, or the directory where it creates arcmail bundles. (What is happening is the mailer is exporting file attach messages without a full path to the files so Rex can't locate the files to send.)

Configuring Beemail

Beemail can operate in both *.MSG and Binkley style outbound mode. If you're using Beemail's Binkley style outbound, enter it in the *Outbound directory* field. See the entry on configuring Binkley style mailers for more information on setting up additional domains.

Configuring FrontDoor APX

FrontDoor/APX doesn't support reading or writing to a *.MSG area, so the *Netmail directory* field isn't available. Instead, Rex needs to know the directory you installed FD/APX to (*FD/APX directory*) and the directory it receives files in (*FD/APX file directory*). If you installed FD/APX to `c:\fdapx`, the default file directory would be `c:\fdapx\file`, but FD/APX allows you to choose an alternate directory so Rex needs to know which one you used.

Address manager

Before sending any mail, Rex needs to know a bit about the netmail addresses your BBS is using and the various zones and domains they're in; these are all configured in the *Address manager* off the main menu.

Getting started

System addresses

In this screen you setup your BBS's netmail addresses. Enter your board's main address in the *Main address* and any secondary addresses in the *System AKAs* list. You can enter up to 100 secondary addresses.

It is important that you enter the right main address. The zone of your main address will be used when there is nothing present to indicate a zone otherwise. (This can happen with some netmail messages, and will be used as the zone of the default outbound directory in Binkley style outbounds.)

Zone matching

This screen helps Rex determine what address to use for which zone. For each zone you think you're likely to send or receive mail, enter an address for Rex to use. For example, if you were a member of Fidonet, you'd enter your Fido address for use with zones 1 through 6.

Domains

Internet Rex fully supports 5D addressing in mailers which understand it. For the moment, this doesn't cover a lot of mailers (AdeptX is the only truly 5D mailer available currently): if your mailer doesn't seem to support domains, it's best to leave this alone. Otherwise, this screen is used to enter the domains to use with various zones. For instance, Fidonet's domain is "fidonet" and its zones are 1 through 6, so an entry should be made for each of the six zones giving the domain as "fidonet".

BinkleyTerm users can also use this screen to enter additional outbound directories. For instance, if your outbound included `c:\binkley\mynet.033`, you would enter zone 51 and domain mynet (remember that the zone extensions in Binkley use hexadecimal numbers). If you don't want to figure out the right zone for your net, you can also specify the zone as 0 and let Rex figure it out for itself.

Email and FTP setup

If you plan on connecting to nodes using email or FTP, you need to tell Rex a bit about what email addresses you use or how you want to use Rex's FTP capabilities.

FTP setup

Setting up FTP for use with Rex is pretty easy: most users won't need to change any of the settings, but just in case, help for them is provided here. Configuring Rex's part in FTP connections is done under the *FTP setup* menu.

Passive mode controls how Rex will try to download files from FTP sites. Most of the time, Rex is smart enough to be able to figure what this setting needs to be for each

Getting started

connection it tries and will set it accordingly for each site it connects to. However, if you know for certain that your machine is behind a firewall or proxy, or is not normally directly reachable from the internet, then you'll find that Rex is setting passive mode on for each connection it makes. This requires a little extra negotiation at start-up: in order to have Rex skip this, you can just turn on passive mode to start off with and Rex will try that type of connection first. The only disadvantage to this is that some FTP sites may not support passive mode: although these are few and far between these days, there are still a few out there.

The *Lock timeout* setting controls how long Rex will wait for FTP locks to clear. If you setup a connection to use locking (see [Using Rex with your own FTP server](#) (Toss to/from a directory) or [Uploading and downloading from an FTP server](#) (Toss to/from an FTP site) for more details on this), you'll generally want Rex to give up on locks after a certain period of time: this lets you specify how long that period of time will be. You can disable this feature entirely by setting the timeout to 0 minutes: for most people, a setting of 5 minutes will work fine.

Related to the Lock timeout field is the *Stale lock timeout* field. Set this field to a non-zero value and Rex will check the date on lock files before processing them. If they're older than the amount of time you give here, it's assumed the locks are stale and Rex will try to remove them.

Finally, the *FTP restarts* setting lets Rex try to resume downloads that might have been aborted in previous connections to an FTP site. With this setting on, download times from FTP sites can drop because Rex only has to download part of aborted downloads, not the whole file. This setting can also result in more hard drive usage, as Rex has to store the partially completed downloads on the hard drive while it's waiting for the next download attempt.

Getting started

Email setup

Because this is one of the most common ways of transferring files and is available to almost everyone with a connection to the net, Rex's set of features for email support is quite rich, and rather detailed. Don't worry if many of the options presented here don't apply to you.

When you choose *Email setup*, you'll be presented with a list of the email addresses configured in Rex. The first time you try this, the list will be empty: hit insert to add a new entry, and you'll be presented with the following menu.

The screenshot shows a text-based menu for configuring an email address. At the top, a header box contains the text: "Internet Rex version 2.00 (Win95/WinNT 32-bit)", "(C) Copyright 1997-2000 Khan Software", and "Press F1 for help". Below this is a window titled "Email address configuration" with a cursor at the beginning. The window contains the following text:

```
Address ID      rex@somewhere.com
User name      rex
Domain name    somewhere.com

Mail spool type  Connect to SMTP/POP3 host
Configure

Domain controlled here  No
Multiuser address      No

Messages downloaded    Matching messages only
```

At the bottom of the window is a button labeled "Quit program".

This is where you configure the email address Rex will be using to send your mail. Addresses you enter here should belong to you: don't enter the email addresses of the people you're sending mail to here! (That's done when you go to enter them into the node manager.) Rex uses the information here to get mail from your servers and identify mail that it sends as coming from you.

You can use any email address you own with Rex - provided the information you enter in the node manager correctly identifies the mail you want to download, Rex won't touch other messages in your mailbox unless you specifically tell it to. For the moment, let's say that the email address you want to add to Rex is `rex@somewhere.com`.

The first thing you're asked to enter, *Address ID*, is just an ID that you'll be giving your email address so you can identify it elsewhere in Rex. Most of the time, the address itself will do, though if you have multiple entries for the same address, you may want to give a little extra information to identify each one.

Email addresses can be broken down into a username and a domain name: the break happens at the @ sign. For the address we're using now, that means that the username is `rex` and the domain name is `somewhere.com`. Enter these values into the *User name* and *Domain name* fields.

Getting started

To control how Rex will be using your mailbox, you have to set the *Messages downloaded* field. This field controls what messages Rex will download from your mailbox. Setting it to “Matching messages only” means that it will only download messages which fit the matching rules you setup for nodes in the node manager or gateway setup, leaving all other messages in the mailbox for you to deal with later. This will fit the needs of most people. If you think you’ll be using this email address to receive mail through Rex from people that you might not have listed in the node manager, you can have Rex download mail from certain programs (other internet transport programs such as Fido2Int, TransX and Internet Rex are listed as options).

The last setting to be configured is the most complex: the *Mail spool type*. There are a lot of different programs available for transferring mail to and from the internet, and Rex can talk to a large number of these, or alternatively, it can talk directly with the servers providing the mail service in the first place. Choose the mail spool type which best fits your way of sending mail. If you’re not sure which method applies to you, check with the people providing your email address and they should be able to tell you the information you need. Most people will find they can use “Connect to POP3/SMTP host” setting.

Configuring a POP3/SMTP connection

If you use a POP3/SMTP server to transfer your mail, your provider should already have given you the machine names or IP addresses of the POP3 and SMTP servers you should be using: enter these into the *POP3 server* and *SMTP server* fields. If you have only one server listed for mail service, enter it into both fields. Some ISPs require their users to connect to the POP3 server before they’re allowed to use the SMTP server: if that’s the case with your ISP, set *SMTP must POP first* to “Yes”. Others require that their users log in to the SMTP server before being allowed to send mail. If yours does this, enter the username and password for your SMTP server in the *SMTP username* and *SMTP password* fields.

To pick up mail, Rex also needs your *POP3 username* and *POP3 password*. Most of the time, the username will be the one in your email address. (For our example so far, that would mean the POP3 username would be “rex”.) The POP3 password is the password you use to pick up your mail: your email provider should be able to give this to you if you don’t know it.

The last setting you may need to set is whether to *Use APOP* to login or not. APOP logins are a more advanced login sequence which send a special code to login to the POP3 server instead of sending your username and password across the internet “in the clear”. This keeps people who might be watching your connection from seeing your email password and getting hold of your mail. Not all POP3 servers support this login; not all POP3 servers support the standard username/password login either. Rex will try to guess the right method to use when logging in, but may not always be successful. If you find Rex giving errors while trying to log in to your mail server and you’re sure your password is correct, try switching this setting.

The remaining settings in this menu are values you may want to tune after having run Rex for a while. If you find you are receiving so much mail for Rex that new messages appear between the time you start downloading mail and the time Rex is done downloading it, you should set *Refetch mail* to “yes”. This will tell Rex to log back in to your mail server to check

Getting started

for new messages after it's done downloading the ones already there. The second field, *POP3 timelimit*, is also related to having a lot of mail to download. If your connection to the internet is time limited (i.e. you can only connect for 10 minutes at a time before you're disconnected by your provider), and it often takes longer than that for Rex to download all your mail, you should set *POP3 timelimit* to "yes" and enter a time limit a little less than the one for your connection to the net in the *Synch every* field. This will tell Rex to log out of your mail server after the time in *Synch every* has passed and log back in again if it hasn't finished downloading all the mail. The reason for doing this is that if a connection to a mail server is dropped while Rex is picking up mail, the messages Rex told it to delete won't be deleted, and Rex will end up downloading them again the next time it connects.

Configuring a UUCP connection

For a UUCP connection, Rex needs you to specify the full path to your UUCP spool in *UUCP path*, the site name for your machine in *Site name* and your host's host name in *Host name*. You can find the site name and host name for your setup in the domain your machine has for email addresses. For instance, if you had the rex.somewhere.com domain, your site name would likely be "rex" and your host name would be "somewhere.com". You'll also have to tell Rex what sort of UUCP program you're using to transfer your mail by setting the *Spool type* field. Most DOS, OS/2 and Windows UUCP transfer programs use either the "UUCICO DOS filename" or "UUCICO DOS filename (no bitmask)" spools, whereas UNIX UUCP programs will likely use the "UNIX long filenames" spool type. At present, Rex doesn't support the UUPC spool type.

You can also configure the grade of mail Rex will generate by setting the *UUCP grade* field.

Configuring a connection with a Fido to Internet gateway

Rex supports two types of Fido to internet gateway connections: one where messages have to be sent individually (set *Mail spool type* to "Gateway") and one where messages can be put into a .PKT by Rex (set *Mail spool type* to "Gateway with packets"). In either connection, Rex requires that you give your *Gateway's address*: this should be the Fido style address to send internet messages to. The flags the messages are sent with can be set in the *Message flags* field: you can scroll through these choices to set any combination of "Kill/sent", "Direct" or "Crash". Finally, because internet addresses don't always fit in the To: line of standard Fido messages, most gateways support two types of addressing: standard and UUCP style addressing. Standard addressing tries to fit the email address of the person you're sending mail to in the To: line of the message. If it doesn't fit, it resorts to UUCP style addressing, where the To: line just contains "UUCP" and Rex adds a line to the top of the message which says "To: email@address". Turning on UUCP addressing uses this style all the time: leaving it off will try the standard addressing first.

If you can send packets to your gateway and you chose "Gateway with packets" as the *Mail spool type*, there are a few extra fields to set. The *Packet path* is the drive and directory where Rex should create the packets. The *Packet password* is the packet password you might

Getting started

have setup with your gateway; if there isn't one, just leave this blank. Finally, the *Packet type* controls what sort of packet to create: "Type 2+" should work fine for most gateways.

Configuring KA9Q style SMTP spools

This type of mail pool only needs two pieces of information: the directory where your inbound mail is saved (*SMTP inbound path*) and the directory where outbound mail should be written (*SMTP outbound path*). Both directories should contain *.TXT and *.WRK files.

Configuring Soup/Yarn mail spools

Soup or Yarn compatible mail spools store all the inbound and outbound mail in one directory. You should specify this directory in the *Queue path* field. It should contain a file called AREAS for Rex to read.

Configuring Rex to work with PMMail

Rex can use the email program PMMail to send and receive messages by reading and writing them straight from its mailboxes. The *PMMail inbound* field should be the directory where inbound messages for the email account you chose are stored. Similarly, the *PMMail outbound* is the directory where outbound messages are. Both these directories are usually subdirectories off the directory where you installed PMMail.

☞ You should be sure not to run Rex at the same time as you are running PMMail, or the mail queues may become corrupt. (This can be fixed by having PMMail rebuild them, but it should be avoided in the first place.)

Configuring Rex to work with the Postroad mailer

To use the Postroad mailer in conjunction with Rex to send and receive email, all you need to do is give the *Queue path* for your email address. Usually this is a subdirectory of your Postroad install path based on your email address's username. For instance, if your email address was cruden@cs.ualberta.ca, and you installed Postroad mailer to c:\Postroad, the queue path would likely be c:\Postroad\cruden.

Getting started

Configuring Rex to work with MR/2 Ice

To work with MR/2 Ice, all Rex needs is for you to enter the path to the directory where MR/2 Ice stores mail in the *Queue path* field. Usually this path is the mail subdirectory of your MR/2 Ice install directory. For instance, if you had installed MR/2 Ice to c:\mr2i, the queue path would be c:\mr2i\mail.

☞ You should be sure not to run Rex at the same time as you are running MR/2 Ice, or the mail queues may become corrupt. (This can be fixed by having MR/2 Ice rebuild them, but it should be avoided in the first place.)

Configuring Rex to work with Eudora

Rex can read and write to Eudora's mailbox files so it sends and receives mail for Rex. To do this, the *Queue path* needs to be set to the directory where Eudora keeps its inbound and outbound mailboxes; the directory should contain the files IN.TOC and OUT.TOC.

☞ You should be sure not to run Rex at the same time as you are running Eudora, or the mail queues may become corrupt. (This can be fixed by having Eudora rebuild them, but it should be avoided in the first place.)

Configuring Rex to work with Nettamer

To read and write Nettamer's mail queue, all Rex needs is the path to the directory where you installed Nettamer, specified in *Queue path*.

Configuring Rex to work with itself (Rex native)

If you decide to let Rex run as the POP3 and SMTP daemon on your computer, you can have your mail delivered straight to the server without going through the hoops of logging in and out. All you need to do is specify the *Username* of the email address to read messages from.

☞ Rex native mailboxes can only be used if the mail daemons and program sending/receiving the messages are one and the same: not just the same executable, but the same instance of the program. Starting up Rex in daemon mode in one session and trying to use a Rex native mailbox session in another won't work.

Getting started

Configuring Rex to work with a Unix mail spool

This configuration should be ideal for people running their own mail daemons under Linux. Rex will download messages from a standard Unix mail spool file and send messages through an SMTP server that you provide. To do this, it needs you to provide the full path and filename of the *Mailbox file* containing the mail to download, and the name or IP address of your *SMTP server*.

Connecting to the net

With the basic information about your system now entered into Internet Rex, configuration is almost complete. If you intend to use Internet Rex over a dialup connection to the internet, then you'll need to configure that connection in the *Connecting to the net* screen. This is only necessary if:

- You intend to transfer mail by connecting to an FTP or BinkP server, or you intend to transfer mail over email and you want Rex to talk directly to an SMTP or POP3 server.
- You are using a dialup connection to the internet or you are using the DOS version of Internet Rex.

If you use an external program to transfer mail (such as UUCICO for a UUCP connection, KA9Q, Soup/Yarn or even commercial mail programs like Eudora, PMMail or MR/2 Ice), these have their own built-in methods of connecting to the net. Similarly, if you're just going to run Rex to move files for an FTP server, you don't actually need to be connected to the net. Finally, if your connection to the internet is over a local area network (LAN), you should set that up under the operating system itself: Rex will be able to hook into that directly without any need for configuration. (This isn't true of DOS, where TCP/IP connections are tacked onto the operating system.) However, most people's connection to the internet run over a dialup connection, and most will fall into one of the two categories listed above.

Each operating system Rex runs under has a different way of setting up connections to the internet. All four operating systems have two fields in common though: *Socket timeout* and *Data timeout*. These control how long Rex will wait for connections to be established (*Socket timeout*) and how long it will wait for connections to be dropped because of inactivity (*Data timeout*). Higher values will work better on slow or poor connections: adjust these accordingly if you find connections timing out when you know they should be working.

Getting started

Connecting to the net under Linux

Internet Rex for Linux assumes that you'll have a batch file available to connect you to the net automatically. When you set *Use dialup networking* to "yes", Rex will call the batch file `dial` to setup a connection to the internet. When the batch file returns, Rex will check its semaphore directory (*Files, paths and programs* → *Internet Rex files and paths* → *Semaphore directory*) for the existence of the file `noconn`. If that file isn't there, Rex will proceed assuming it now has a connection to the net: otherwise, it will assume the connection failed and won't try to contact any internet sites.

If the connection was successful, Rex will call the batch file `hangup` when it's done to terminate the connection to the net.

Connecting to the net under DOS

Connecting to the internet under DOS has two categories of information to setup, one for standard network information, and an additional category of information for those using dialup connections. If your connection to the net is over a LAN, skip the dialup configuration section.

Configuring dialup information

Internet Rex for DOS includes everything you'll need to establish a connection to the net over a dialup link using PPP. Setting this up requires filling in information about your modem and the logging at your provider's end, and then writing a connection script to start everything up.

The first part is configuring the modem for use with Rex. Fill in your modem's *Init string* and tell Rex what *COM port* to use. Rex will fill in the modem's *Address* and *IRQ* from the COM port information you give it; if you're using a non-standard modem address or IRQ, you should change the values Rex fills in so they're correct. You should also specify the *Baud* rate your computer talks to your modem at.

☞ Note that the *Baud* field doesn't refer to the top speed your modem runs at (e.g. 28800, 33600 or 57600), but rather the speed at which your computer talks to your modem. (Users of a FOSSIL driver will recognise this as the locked baud rate.) Usually this is 38400 or 57600: Rex can accept values as high as 115200 for this value.

Then fill in the *Login name* and *Password* you use to connect to your internet provider, and the *Phone number* to dial to connect to them. Finally, give the number of times to retry the connection if the provider is busy in *Retries*: zero means keep trying until connected.

With the information you've given, Rex will create a sample script called `rex.scr` for use with the Rxdial dialler, a configuration file for the PPP driver called `pppd.cfg`, and a batch file called `dial.bat` to run everything. You should read the documentation provided for Rxdial (in `rexdial.doc`) and tune the script Rex created so that it logs on to your provider correctly. You will also need to configure the network information for your connection to the internet.

Getting started

Configuring network information

DOS generally doesn't provide many ways for finding out information about the network connection to the internet, so you may have to specify some of the information here. Your internet provider should be able to supply you with the values for these fields if you're not sure what they are. If your packet driver supports the BootP protocol for auto-configuring network connections, you can just enable that by setting *Use bootp* to "yes" and skip the rest of the setup.

The *IP address* field should contain the internet address of your machine. Many dialup connections and some network connections provide different internet addresses each time the machine connects to the internet: if this is the case with your connection, enter "0.0.0.0" as your internet address. This will tell Rex to try to figure out what the internet address is each time it is run.

The *Network mask* field tells Rex how large a network you're running on. For most connections this will be "255.255.255.0". Your provider should have supplied you with the netmask for your connection.

To translate internet machine names like ftp.somewhere.com to an IP address which machines can understand, internet programs must connect to a name server. You can specify the address of up to two name servers for Rex to use: you must give at least one.

You can also adjust Rex's network speed settings by setting the *MSS* setting. For ethernet connections, a larger MSS (around 1400) will provide more efficient transfers. For slow connections like modems, a smaller MSS (around 256) lets the connection deal better with overruns that are common coming from a fast connection to the slow modem.

Connecting to the net under Windows 95/98/NT

Windows fortunately provides built in dialup networking software for connecting your computer to the net, and Rex can take full advantage of that, down to running scripts, dialling and hanging up.

Dialup networking either turns on or turns off Rex's use of Windows's dialup networking feature. If you have a connection to the net over a LAN, leave this off and Rex will use that connection. If you want Rex to use one of the connections you've defined in Windows's dialup networking setup, turn this on. The remaining fields apply only to dialup connections.

You can control whether Rex should automatically try to dial out or hang-up connections using the *Auto-dial* and *Auto-hangup* fields. When Rex is set to auto-dial, if a connection to the net hasn't been established using the connection you specify in *Connection*, Rex will start dialling your provider and invoking whatever scripts are needed to get connected. It will dial as many times as you specify in the *Redials* field before giving up. Once Rex is done, if you've turned on auto-hangup, Rex will hang up the phone and stop the connection, assuming it's the only program left using it.

By default, Rex will try to use the *Connection* you've told it to, but if you've connected to the net through a different dialup networking connection, it needs to know how to behave.

Getting started

If you can reach all the sites you need to through any of your dialup networking connections, you can set *Other connections* to “Use others if available”, and Rex will use whatever connection has been established. Some providers limit access to their mail servers to those who have connected through their dialup modems. In cases like these, you should leave *Other connections* at “Use this one only”. Rex will then only connect to internet sites if it can establish a connection over the connection you gave: if it can’t do that, it will abort.

Besides specifying the connection to use, you can also tell Rex what *Login name* and *Password* to use when calling your provider by filling in the appropriate fields. Finally, the *Inactivity timeout* field can be used to specify the inactivity timeout for your connection. Many providers have timeouts, where if no data is sent for a certain amount of time the connection is automatically dropped. Since Rex may spend some time doing online processing, you can set Rex up to send data over the line every few seconds to prevent your provider from dropping the connection. How often this data is sent is specified here: if you don’t want to use this feature, just set the timeout to 0 seconds.

Almost all of the options presented in this menu can also be controlled from the command line. See the section on [The command line](#) for more information on this.

Connecting to the net in OS/2

Installing an automatic dialup connection to the net in OS/2 is quite similar to the procedure for connecting to the net in DOS automatically. Although there are a number of excellent OS/2 programs for automatically dialling and connecting to an internet provider (InJoy in particular is recommended for this purpose), Rex does also include its own scriptable dialler which can be used along with the PPP.EXE and SLIP.EXE programs included with OS/2’s connection packages to provide a reasonably good automatic connection to the net.

In order to do this, Rex needs a bit of information about your modem and the way you’ll be connecting to the net, namely your modem’s COM port, the init string to use to initialise it and the baud rate to talk to it at. You can enter these in the *COM port*, *Init string* and *Baud rate* fields.

☞ Note that the *Baud rate* field doesn’t refer to the top speed your modem runs at (e.g. 28800, 33600 or 57600), but rather the speed with which your computer talks to your modem. (Users of a FOSSIL driver will recognise this as the locked baud rate.) Usually this is 38400 or 57600; the highest value Rex will accept for this field is 57600. (This is a limitation of the COM drivers included with OS/2.)

The *Redials* and *Phone number* fields tell Rex what number to call to establish the internet connection and how many times to retry dialling in case there isn’t a connect on the first try. Once connected, the *Username* and *Password* fields are used to create a rough script that will be used to log in. The actual script generated may not work with your particular login sequence: you should examine the script that Rex creates (in *rex.scr*) and modify it to suit your needs. (Refer to the documentation on RexdialP included with Rex for help with configuring the dialler.)

All this information is combined into a batch file that calls the dialler first, then one of OS/2’s connection establishment programs (either SLIP.EXE or PPP.EXE), then Rex. Which

Getting started

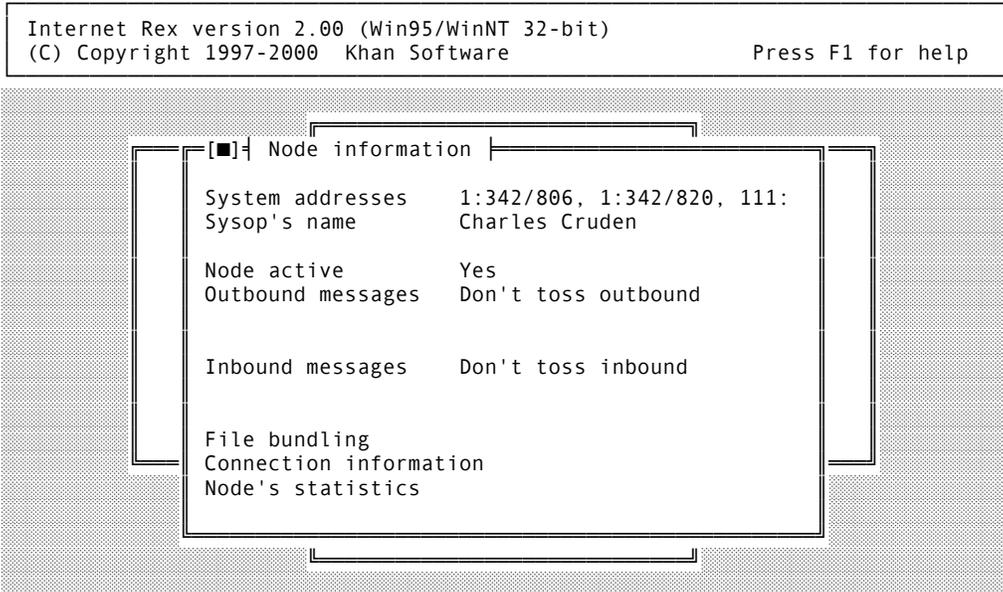
establishment program is called is determined by the *SLIP or PPP* setting. (Note that setting this to “SLIP” will require you to specify your machine’s IP address and your uplink’s address in the *Your IP address* and *Provider’s IP address* fields.) When you set *Use dialup networking* to “yes”, Rex will call this batch file (`dial.cmd`) to connect to the net. After running the batch file, Rex checks its semaphore directory (*Files, paths and programs* → *Internet Rex files and paths* → *Semaphore directory*) for the existence of the file `noconn`. If that file is there, it assumes the batch file couldn’t establish a connection to the net: otherwise, it proceeds as usual. When it’s done, Rex runs the batch file `hangup.cmd` to shut down the connection and hang up the phone.

Inactivity timeout provides you with some extra control over the connection once it’s been established. Some providers have built in timeouts for the connections they support: if no data is sent for a certain amount of time, the link is dropped. You can get around this happening while Rex is running using the *Inactivity timeout* field: setting this to 60 seconds, for instance, would tell Rex to send data over the modem every 60 seconds, regardless of what it’s doing.

Setting up a node

The main reason for Internet Rex's existence is to send your BBS mail out over the net for your connections to receive. As such, the *Node manager* is probably the most important menu in Rex's setup program; it's here you add or remove connections telling Rex who to send mail to and how.

After selecting *Node manager* from the main menu for the first time, you'll be presented with a blank menu where connections are normally listed. Hit the `insert` key to add a new connection and you'll be presented with this menu.



The first few fields are what you'd expect. The *Sysop's name* field is the name of the sysop running this system. The *System addresses* are the netmail addresses of the system you want Rex to send mail to (e.g. 1:342/806, 1:342/820, 169:4100/101). You can enter up to two hundred system addresses for each node: Rex will treat mail to any of the addresses you give as being for this node. You can also route mail for other addresses through a given node using the routing control file `route.rex`. Check the Routing section of this manual for more information about how to setup routing in Rex.

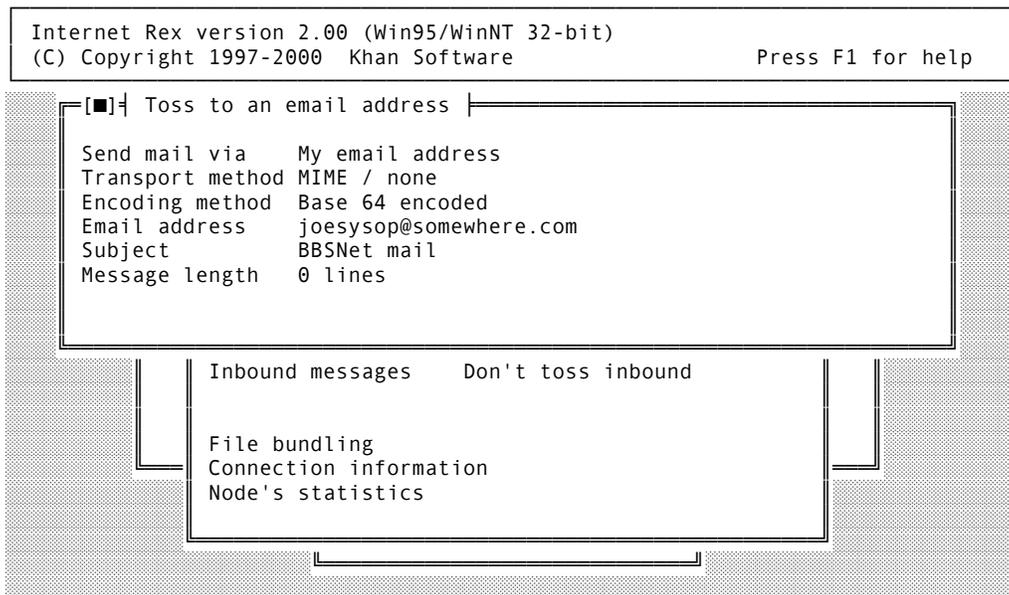
A number of different ways for sending BBS mail over the internet have evolved in the short time the net has been around. Rex supports four different methods for doing this right now: connecting to a BinkP server, uploading and downloading files from an FTP server, moving files to directories on your own FTP server and sending and receiving messages through email. Rex lets you mix and match each of these methods for inbound and outbound mail for a given node. Each method will be covered as a pair though, as this is the usual combination to use.

Setting up a node

Setting up an email link (Toss to/from mail)

Sending mail

This is probably the single most common use for Rex - sending and receiving BBS mail over email. To setup a connection over email, you'll need the email address of the person you're sending to and at least some idea of what encoding they want the mail sent with. If you're not sure what encoding to use, find out what program they use to send/receive their BBS email. To set a node up for sending mail over email, set the *Outbound messages* field to "Toss to mail" and choose *Configure* just below it. You'll be presented with this menu:



If you have multiple email addresses setup in the *Email setup* menu of Rex, the *Send mail via* field will be blank: hit enter to select which email address you'll be using to send mail to this node.

The *Transport method* and *Encoding method* fields determine what sort of email message Rex will create to send mail to this node. This is where it's important that you know what sort of encoding you wanted to use, or what program the person on the other end is using. Here's a table of commonly encountered programs, suggested transport methods and encoding methods:

Setting up a node

Program remote is using	Transport method	Encoding method	Additional information
Internet Rex >2.00	SEAT	Rex encoded	To use Rex encoding, both ends of the connection must be using Rex 2.00 or higher
Internet Rex <1.19	SEAT	UUencoded	
TransX	TransX compatible		
Fidonet2Internet mailer	MIME/none	Base64	
Allfix	Allfix		
PXFids	FIDS		
Watergate	MIME/none	UUencoded	
GIGO	MIME/none	UUencoded	
TransNet	MIME/none	Base64	Requires special subject line
SEAT compatible mailers	SEAT	UUencoded	
Most email readers (Eudora, PMMail, Nettamer, etc.)	MIME/none	Base64	

If you're not sure what program the person you're sending mail to is using, your best choice for a transport method is MIME/none. Similarly, if you're not sure what encoding method to use, your best bet is Base64.

Unlike some programs, Rex allows you to specify the subject line for mail you send to a node. You can enter that in the *Subject* field. Note that some transport methods require a particular subject: you can't set the subject for TransX transport nodes. Also, if you are sending mail to someone using TransNet, you must set the subject to "Transport From-(address) To-(address) (password)", replacing the values in brackets appropriately. (For instance, if the address you were using for this node were 1:342/806 and the address for the remote were 1:342/820, and you had a password of "Blah", the subject should be "Transport From-1:342/806 To-1:342/820 Blah".)

The last field (second last for TransX nodes) is *Chunk size*. This lets you control the size of the chunks Rex will split large files into. Many email servers don't allow messages above a particular size to be sent or received, or if they allow large messages, start to split them themselves. This message splitting can make messages unreadable by some programs when they're received. You should find out if your mail server splits large messages and if it does, how large messages are allowed to get. Depending on the transport method you chose, you can set Rex up to split messages every so many lines or every so many bytes. If you don't want Rex to split messages for this node, you can just set this value to zero.

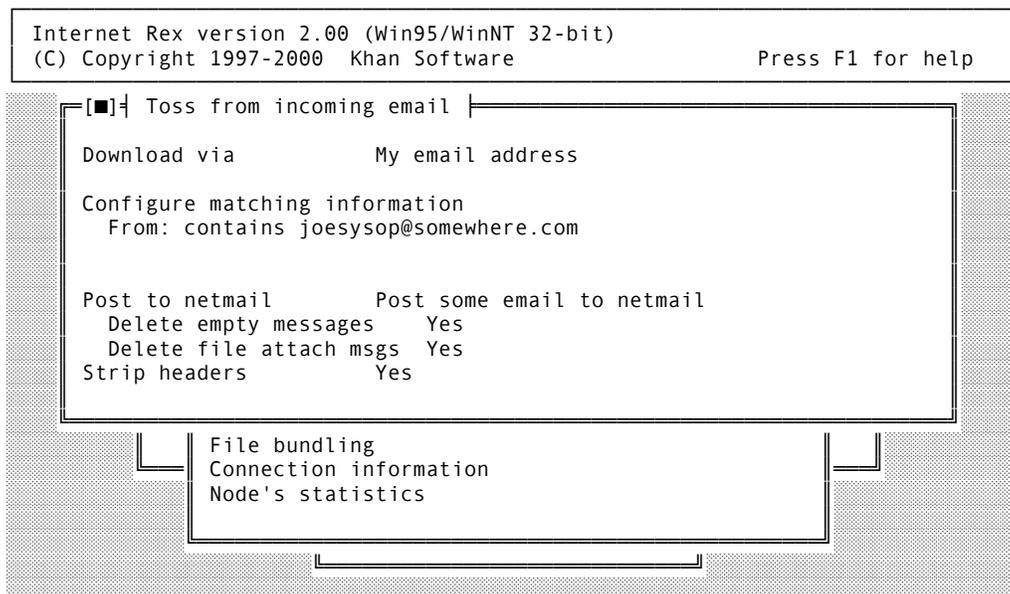
Setting up a node

- ☞ Some transport methods force you to split messages at a certain size. Allfix requires that files be split into chunks at most 16K large, and TransX requires that chunks be at most 60K in size.

TransX connections will have one last field, a *Session password*. This is the security password you can add to TransX mail to authenticate the message. If you have a session password setup with the remote node, enter it here; if you don't have a password, just leave the field blank.

Receiving mail

Now that the sending part of the connection is setup, we can proceed to the receiving part. To set this up, go back to the main node editor menu and set the *Inbound messages* field to "Toss from mail" and choose *Configure*. You'll be presented with this menu:



If you just configured the "Toss to mail" part of Rex, most of this screen will have been setup for you already. The *Download via* field should be set to the email address (or email addresses if you have more than one) you gave this node to send you mail at. Usually, this is the same as the email address you use to send mail to this node.

The matching information field is probably the trickiest part of configuring Rex to receive mail from someone. This field should contain a list of rules Rex will use to determine if a given email message is from this node. The format is a bit like a web search engine: you choose a field to search and a value to search for in that field, and optionally, a conjunction to add to the end of it to restrict or widen your search a bit. The usual first rule to enter is "From contains *user's email address*". This tells Rex that messages with a From: containing the user's email address are from this user. If you wanted Rex to only process messages from this user whose subject was "BBS mail", you would edit the first rule so it read "From contains *user's email address* and" and add in a second rule: "Subject contains BBS mail". If the user was sending you messages from two different email addresses, you could change the rules to

Setting up a node

read “From contains *email address #1* or” and “From contains *email address #2*”. The important thing to remember is make sure the rules you give will match *all* the mail you want Rex to download from this user, and *only* mail from this user.

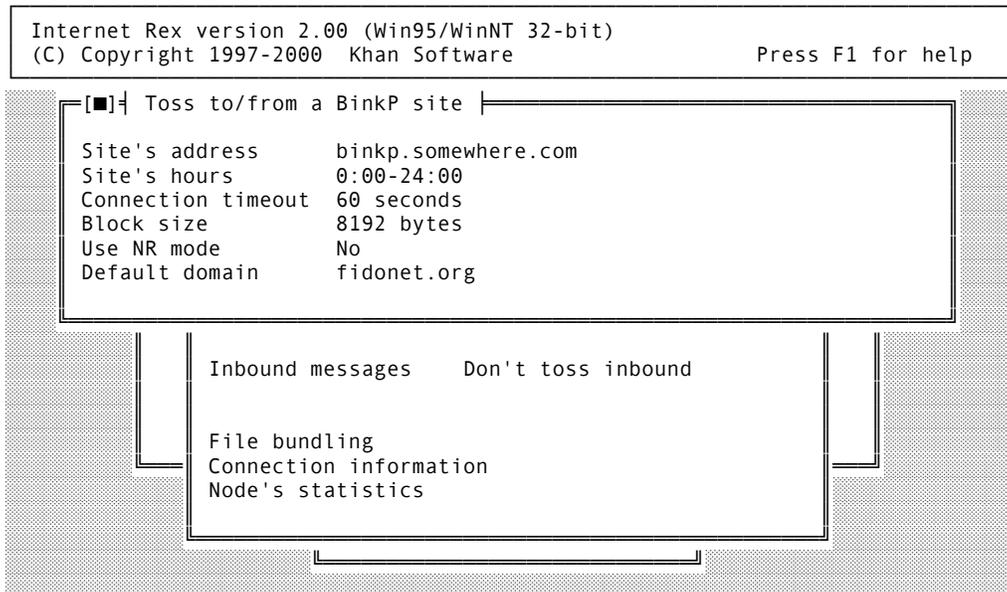
Finally, there is the *Post to netmail* field. This controls what Rex will do with the email messages it's received from this node once they're downloaded. “Do not post email to netmail” will make Rex decode whatever files are attached to the email message and then simply delete the message. “Post some email to netmail” means that Rex will post some of the email messages it downloads to netmail: you can control which ones by setting the *Delete empty messages* and *Delete file attach messages* fields. If *Delete empty messages* is set to yes, Rex will delete any message where there is no text for you to read (e.g. messages which contain only a file attached to them, or some messages from Allfix which have headers but no text). *Delete file attach messages* tells Rex to delete any message which has a file encoded in it. The remaining email messages Rex receives from this node will be posted in netmail messages addressed to you. “Post all email to netmail” just shuttles every email message Rex downloads from this node into a netmail message, with files decoded and such.

When Rex does post a message to your netmail folder, you can control whether or not the message headers (the From:, To:, Subject: Received-By:, etc. stuff at the top of messages) are posted along with the message by setting the *Strip headers* to “yes” or “no” as you like: “yes” will nuke the headers, “no” will leave them in.

Connecting to a BinkP server (Toss to/from a BinkP site)

Whether you're connecting to a BinkP server to pick up or send mail, the settings in Rex will be the same. You first set *Inbound messages* or *Outbound messages* to “Toss to/from a BinkP site”, and then choose *Configure* to setup information about the site. You'll be presented with this menu:

Setting up a node



The site you're connecting to should have provided you with his internet address and the hours the server is available; enter these in the *Site's address* and *Site's hours* fields. The remaining values here determine how Rex will talk with the server.

The *Connection timeout* lets you set how long Rex will wait for data from the site before deciding a problem has occurred and dropping the connection. With any luck, this will hardly ever happen, but if you find a site is repeatedly timing out, you may want to try upping this value. The *Block size* lets you set the size of the chunks the protocol will break files into. This is a bit like Zmodem's chunk size in that the bigger the chunks are, the more efficient the protocol is; larger chunks also mean more data has to be resent if there's a problem with the sending. You can specify block sizes from 1K to 32K.

Connections to BinkP sites can be unreliable at times, with connection losses or garbage data. If you find your BinkP connection is not working well, try setting *Use NR mode* to "yes". This will enable non-reliable BinkP connections for this node. You should only use this if your connection is poor: enabling it on a good connection will actually degrade performance.

The *Default domain* is used when Rex is connecting to the BinkP site. BinkP is a purely 5D protocol: that means that any addresses that get presented to the remote site have to have a domain as well as the usual zone, net, node and point. Since most people won't have domains for their all their addresses (if any), Rex needs a dummy domain to add to the end of your addresses when it sends them to the BinkP site, and that's where the default domain comes in. Note that this can't just be any old domain though: it has to match the default domain on the server, at least for the addresses you're interested in picking up mail from.

There is one other setting that BinkP sites may setup: a session password. This password is used to authenticate the person connecting to the site, to make sure that mail isn't sent to the wrong person. If you've setup a session password with your site, you can tell Rex what password to use in the *Session password* field of the *Connection information* menu: see the [Secure and encrypted mail](#) section of this manual.

Setting up a node

Using Rex with your own FTP server (Toss to/from a directory)

If you run your own FTP server for mail purposes, Rex can toss mail to and from directories on your hard drive to work alongside your FTP server. To have Rex use this method for a certain node, set the *Inbound messages* or *Outbound messages* to “Toss from directory” or “Toss to directory”, and choose *Configure* underneath. You’ll be presented with a screen something like this:

```
Internet Rex version 1.00 (Win95/WinNT 32-bit)
(C) Copyright 1997, 1998 Khan Software                               Press F1 for help

[■] Toss to a directory

Destination      c:\ftp\users\joe\outbound
Check for locks  Yes
Lock filename(s) c:\ftp\users\joe\*.
Wait for lock    Yes
Create locks     Yes
Lock filename    c:\ftp\users\joe\busy.flg

File bundling
Connection information
Node's statistics
```

The first field when tossing to a directory is *Destination directory*, when tossing from a directory it's *Source directory*: either way, this is the directory on your hard drive you want Rex to move files to, or move files from. It must be a full pathname.

For a lot of people this is all that needs to be setup. There are more options available though, for those who run a system that gets a lot of activity, or where Rex is run quite often. In situations like these, Rex will sometimes be running at the same time that someone is online trying to upload or download files. If this happens, it's possible that Rex will try to move a file someone is in the process of uploading, or toss files into their download directory after they've downloaded, but before they've started deleting the downloads from the server. In this case, you might end up losing files: Rex provides a solution to the concurrency problem by allowing FTP users to have locks setup to show they're online.

A lock is just a file on the hard drive that Rex checks for before trying to process files for a given node: if the file is already there, that means the person is online, and Rex will be careful about processing files for this node. You can also have Rex create locks itself to tell the person connecting to your server that Rex is running, and they shouldn't upload or download stuff just yet. All this is controlled by the *Check for locks* and *Create locks* fields.

Setting *Check for locks* to “yes” will get Rex to check for lock files before processing this node's files. When you set the field to “yes”, two other fields appear: *Lock filename(s)* and *Wait to clear*. In the *Lock filename(s)* field, you specify a filename or wildcard that you want Rex to check for. The *Wait to clear* field tells Rex whether to wait until the file has been removed before proceeding, or just skip sending files at this point entirely: “yes” gets Rex to

Setting up a node

wait, “no” makes it skip. When waiting for lock files to clear, Rex will only wait as long as is specified in the *Lock timeout* field in the *FTP setup* menu.

Setting *Create locks* to “yes” will also bring up another field. *Lock filename* is simply the full name and path of the lock file you want Rex to create when it’s processing files for this node.

If the FTP server you’re setting this node up for is also being run by Rex, consider turning off locking and setting the node up as an auto-toss node. You can find more details about this in the FTP server setup section of this manual under [Adding a user](#).

Uploading and downloading from an FTP server (Toss to/from an FTP site)

To setup a node to send or receive mail via an FTP site, you first need to get the internet address of the FTP site you’ll be transferring files through, as well as the user name and password you’ll be using to log on. The person running the FTP site should have given these to you. They should also have given you the directories that you’ll be uploading or downloading files from. To enter this information into Rex, set the *Inbound messages* or *Outbound messages* field in the node editor to “Toss from an FTP site” or “Toss to an FTP site” as appropriate, and select *Configure*. You’ll be presented with a menu something like this:

The screenshot shows a window titled "Internet Rex version 2.00 (Win95/WinNT 32-bit)" with a copyright notice "(C) Copyright 1997-2000 Khan Software" and a prompt "Press F1 for help". The main content area is titled "Toss to an FTP site" and contains the following configuration fields:

Site name	ftp.somewhere.com
User name	joe.sysop
Password	*****
Site's hours	0:00-24:00
Lock file size	1 byte
Unique storage	No
Filename case	Don't care
Script type	Automatic (constructed by Rex)
Dest. directory	/inbound
Directory locking	Do not use directory locking

First off, the *Site name*, *User name* and *Password* fields should be filled in with the FTP site’s internet address, the user name you log on to the site with, and the password you log on with. If the FTP site you’re connecting to is running on a non-standard port (the site administrator should have told you if it is or not), you can specify that port as part of the site’s name. For instance, to connect to ftp.somewhere.com running on port 2100, enter the site name as “ftp.somewhere.com:2100”. You can also specify the site’s operational hours in the *Site’s hours* field. (The range 0:00 to 24:00, i.e. all the time, can be represented not only by turning on all the times in the list, but also by turning them all off.) If the current time falls

Setting up a node

outside the range you specify here, Rex won't try to connect to the FTP site. Finally, the *Lock file size* tells Rex how big to make lock files for this connection: some FTP sites don't like 0 byte lock files, some don't like 1 byte lock files. If one doesn't work, try the other.

The menu for uploading files to an FTP site will now have the fields *Unique storage* and *Filename case*. These determine how Rex will try to store the files on the FTP site. *Unique storage* tells the FTP site not to overwrite files that are already there: if a file is being uploaded that matches a file already on the site, a new name will be given to it and the file upload will continue. If *Unique storage* is turned off, the file will just be overwritten. The *Filename case* determines how the file's name will be sent to the site. Setting this to "don't care" will use whatever the file was called on the hard drive; "upper case" will send the name in all-capitals; "lower case" will send the name in all lower case letters.

Now comes the tricky part, creating a file transfer script. This is the sequence of commands Rex will issue to the FTP site to change to the right directory and transfer files. You can choose to have Rex try to create a script for you by setting *Script type* to "Automatic", or you can create your own script using Rex's FTP commands by setting it to "Manual". If you choose to create your own script, a *Create script* field will appear and you can enter up to 125 FTP commands for Rex to run through while talking to the FTP site. (For more details on what FTP commands Rex supports, see the reference manual's section on FTP commands.) In the menu to upload files to an FTP site, an *Always run* field will also appear. If you set this to "yes", Rex will try to connect to the FTP site and run this script every time you tell Rex to send mail to FTP sites, even if there's no mail waiting for this node. Most scripts don't need this setting, but it can be useful for some of the more advanced scripting commands available.

Automatically generated FTP scripts should work for most connections, but they may need a bit more information to be entered into Rex for them to work. If you set the script type to "Automatic", the first field after the script type will be either *Destination directory* or *Source directory* (depending on whether this is for a node sending or fetching mail). This is simply the directory you'll be storing files or the directory you'll be retrieving files from. It's important to note that these should be **absolute** paths: that is, the string you enter in the field should change to the same directory no matter what directory you're in on the FTP site. An absolute path should look something like `\out`, or `~in`. `out` or `..\in` won't work: they are relative paths, relative to the directory you're in at the time the CD command was issued. You may be able to get the automatically generated scripts to work with relative directories, but it's not recommended for a first attempt.

After choosing the source or destination directory, you'll have the option of setting up locking for this node. As mentioned in the Using Rex with your own FTP server section, sites which are contacted often or on which a lot of processing goes on can have concurrency problems: users and the server may be trying to access files at the same time. A way of preventing this from happening is by using lock files to tell the server or the user that the other is processing files at the moment and they should wait. Rex lets you optionally check and/or create lock files when logging onto the FTP site. Setup the type of locking you want to use by toggling the value in the *Directory locking field*. This will present you with a few more fields to fill in.

The *Lock directory* should specify the absolute path to the directory where locks will be created and searched for. If you've chosen to create locks, the *Create lock* field should be

Setting up a node

filled with the name of the lock file to create. This is just a filename: no directory information is needed. Something like “lock” or “busy.flg” would work well. Similarly, the *Locks to check* field should be filled with a wildcard to match against files in the lock directory. You can give a single file name if you know the lock you’re looking for, like “lock” or “busy.flg”, or you can give wildcards like “*” or “lock*” to match a range of files. If anyone of the files in the lock directory matches the wildcard, the connection will be considered locked. What happens when the connection is locked is determined by the *Wait to clear* flag: set to “yes”, Rex will wait until the file is removed before proceeding. How long it waits is determined by the setting in *FTP setup* → *Lock timeout*. If *Wait to clear* is set to “no”, Rex will simply end the connection to the site and try again another time.

If you find you must use a manually create script, here are a couple of sample scripts to get you started.

```
cd \download
get *
```

This script simply changes to the \download directory and downloads all the files there to the inbound file directory.

```
cd locks
if exist busy.flg
  wait busy.flg
endif
create busy.flg
cd ../upload
put *
cd ../locks
del busy.flg
```

This is a more complicated script that uses lock files. It first changes to the `locks` directory and checks to see if the file `busy.flg` is present. If it is, it waits until it’s been removed. (If the wait is longer than the timeout you specified in the *FTP setup* menu, the script will abort.) Once the file is gone, it creates its own lock file, then changes to the `upload` directory to upload files. The `put` command transfers all the files for this node from Rex’s queue to the FTP site; then the script changes to the `locks` directory again and removes the lock it put there.

Now that you’ve setup some nodes in Rex, it’s time to run it and start sending mail!

Running Rex

Internet Rex has two basic modes of operation. The one users acquainted with versions less than 2.0 will most be familiar with is the command line mode, where Rex is run from the command line with a series of arguments to define which nodes to fetch and send mail for. Rex can also be run in daemon mode: in this mode it operates more like an internet mailer, running 24 hours a day and executing events at pre-defined times. You must be running Rex in daemon mode in order to use its servers.

DOS Internet Rex can be run in daemon mode under DOS, but it doesn't support servers. You must be using OS/2, Windows or Linux to take advantage of Internet Rex's servers.

The command line

Rex's main program is named differently for each operating system it supports.

DOS The executable is `rex.exe` for the 16-bit version and `rexd.exe` for the 32-bit version. If you configured Rex to use a dialup connection to the internet, `Rexcfg` will have created a batch file called `dial.bat` in the same directory you installed Rex to: this batch file can be run with the same arguments as Rex to automatically connect to your provider and run Rex.

OS/2 The executable is `rexp.exe`. Again, if you configured Rex to use a dialup connection to the internet, the configuration program will have created `dial.cmd` in Rex's directory. You can give the same arguments to this command file as you would to Rex and the file will dial up your provider, connect and run Rex with the arguments you gave.

Win The executable is `rexw.exe`.

Linux The executable is `rex1` if you aren't running through a SOCKS4 server, `rex1s` if you are behind a SOCKS4 server.

Many of the options are the same for each operating system though. Rex's syntax is:

```
rex [-fetch [binkp | dir | ftp | mail | news |
[x]anonymous | [x]e<email ID> | [x]<address(s)> |
[x]g<newsgroups> | [x]s<server ID>]...
[-send [binkp | dir | ftp | mail | news |
[x]anonymous | [x]e<email ID> | [x]<address(s)> |
{x]g<newsgroups> | [x]s<server ID>]...
[-runscript <script file> [<script file>...]]
[-map <map label>]
[-processor <0 | 1 | 2 | 3>]
[-trim] [-event <event tag>]
[-maint] [[+|-]between]
[-daemon]
[[+|-]dun [<connection> [<username> [<password>]]]]
[[-help | -notify] [[x]<address(s)>]] [-?]
```

Running Rex

If you run `rex` all by itself (without any additional switches), users running in beginner mode will receive help on the command line options. For users at a higher user level, Rex will fetch and send all mail for all nodes.

-fetch

This option tells Rex what nodes to fetch mail for. Without any qualifiers, Rex fetches all waiting mail for all nodes. By adding qualifiers, Rex instead fetches mail only for the nodes specified. Each of the options select all the nodes using the relevant transfer method to receive their mail, so that `binkp` gets Rex to fetch all mail from BinkP nodes, `dir` from “Toss from directory” nodes, `ftp` works for FTP nodes, `mail` fetches mail from email nodes and gates, and `news` fetches newsgroup messages. `[x]e<email ID>` also applies to email nodes: without the `x` in front, it selects all the nodes and gates that receive mail through the email address with the ID given in `<email ID>`. With the `x` in front, all those nodes are excluded from having their mail fetched. The `[x]anonymous` specifies ‘anonymous’ email nodes: email that Rex can’t identify as being from a particular node, but which it would download because of some other setting (e.g. email FREQs, gated messages, stray Internet Rex messages, etc.). Specifying `mail` automatically includes anonymous nodes: you can exclude them from download when mail is specified by putting an `x` in front of the `anonymous`. Similarly, if `mail` isn’t specified, you can still pick up anonymous mail by giving the `anonymous` switch. `[x]g<newsgroups>` allows you to fetch (or not fetch by specifying the `x`) messages in a newsgroup or set of newsgroups. `<newsgroups>` can be either the name of a single group or a wildcard to match against all your groups. Finally, `[x]s<server ID>` allows you to fetch (or not fetch by specifying the `x`) all the newsgroups associated with the news server whose ID is `<server ID>`.

You can also specify or exclude individual nodes to pick up mail from by adding the address after the fetch command, or putting an `x` in front of the address to exclude it from the list.

`-fetch`

Fetch all mail for all nodes.

`-fetch ftp`

Only pick up mail from nodes whose mail is waiting on FTP sites. Don’t download email, connect to BinkP servers or check directories for mail.

`-fetch ftp dir`

Only pick up mail from nodes whose mail is waiting on FTP sites or in a local directory.

`-fetch ftp x1:342/806`

Picks up mail from FTP sites. If 1:342/806 receives its mail from an FTP site, its mail is *not* picked up.

`-fetch mail xeRex`

Fetches mail for all nodes sending files via email, except those nodes sending you files through the email address you gave the address ID of “Rex”.

`-fetch galt.*`

Fetches news for all the newsgroup gates you’ve setup which match `alt.*`.

Running Rex

-send

This option works exactly like the fetch option, only it's for sending files and mail. See the section on the fetch option for more details. The only difference here is the anonymous mail switch includes any netmail gated to email.

-runscript <script file> [<script file>...]

Use this to have Rex run the FTP script in the file you give. You can specify multiple script files in a list, but none of them can contain wildcards.

-map <map label>

Using this option, you can change the default drive mapping setup in *Files, paths and programs* → *Drive mappings (DOS, Win95, OS/2)* to the mapping with the label you give. This will let you run Rex on machines with different drive mappings without having to run the configuration program to change the mapping.

-processor <0 | 1 | 2 | 3>

If you want to change the *Processor friendliness* of Rex at runtime, you can do so by giving the -processor switch. The number afterwards represents how often Rex will give up the processor to other tasks: the lower the number, the less often it will give it up.

-trim

This option simply tells Rex to trim the log file according to the settings in the *Logging options* screen. (If you don't have log auto-trimming enabled, this is the only way to have Rex trim the log file.)

-event <event tag>

Normally, if Rex is run from the command line, there will be no active event, so any event specific routing commands in the `route.rex` file will be ignored. You can use this command line option to tell Rex to use the routing commands in the event section of `route.rex` that has the event tag you gave.

-maint

Use this to run Rex's midnight maintenance routines manually. This checks the confirmation status of gateway mailing list users and expires or deletes them if necessary. This should only be used if Rex isn't running in daemon mode at midnight.

[+|-]between

This overrides the setting for running the BETWEEN.BAT batch file in *Files, paths and programs* → *BETWEEN.BAT*. Giving it with a + in front will make Rex run the BETWEEN.BAT file during this run; a - will disable running BETWEEN.BAT.

-daemon

Running Rex with this switch will put it into daemon mode.

Running Rex

[+|-]dun [<connection> [<username> [<password>]]]

This little option lets you override most of the settings in the *Connecting to the net* screen of the configuration program. `-dun` disables dialup networking in Rex: Rex will either use a LAN connection if it's available, or nothing at all.

`+dun` enables dialup networking.

Win Windows users can optionally specify first a different connection to use, then a different username to use, then a different password to use. For instance:

```
rexw +dun ConnectionTwo
```

will tell Rex to use the connection labelled `ConnectionTwo` to connect to the net instead of the usual one. Giving:

```
rexw +dun ConnectionTwo Bubba MYPassWord
```

additionally tells Rex to use the username `Bubba` instead of what's in Rex, and the password `MYPassWord`.

[-help | -notify] [[x]<address(s)>]

These two options allow you to create RexFix help and status messages for the nodes with the addresses you specify afterwards. Specifying `x` in front of an address will exclude that address from whatever list you'd presented so far. Addresses can also be specified as wildcards. For instance:

```
rex -notify 169:*
```

Sends notify messages to everyone in the node manager in zone 169.

```
Rex -help 169:* x169:4??*/*
```

Sends help messages to everyone in the node manager in zone 169, except those in zone 169:4000.

If you don't give any addresses to send to, Rex will send help or notify messages to everyone in the node manager.

-?

Running Rex with just `-?` will bring up a listing of all the command line options in Rex. Giving this option after one of the normal options in Rex will bring up a help screen with help about that option. For example:

```
C:\IREX> rex -trim -?
```

```
-trim: If you specify -trim on the command line, Rex will trim the log file it has generated to the size you specify in Rexcfg. If you've specified a large logfile size, this may take a while to do.
```

e.g. `rex -trim`

```
C:\IREX>
```

Running Rex

Win The Windows version of Rex includes a few command line options which apply only to it.

[+|-]autodial

[+|-]autohangup

These options allow you to override the *Autodial* and *Autohangup* settings in the *Connecting to the net* screen for Windows users. Specifying `+autodial` will tell Rex to dial out to connect to the internet, regardless of how you have this specified in the config program itself. (The same restrictions still apply: Rex won't dial out if the line is in use, or if the connection has already been established.) Specifying `-autodial` has the opposite effect: Rex will not try to dial out to connect to the internet. The same logic applies to the `+/-autohangup` option.

DOS DOS users also have an option available in Rex which is only available to them.

-ip <ip address>

Using this switch, you can override whatever other information Rex might get about your IP address and tell it to use the IP address you gave instead. (This will only work if you haven't enabled auto-configuration through BootP.)

Command line shortcuts

Having to specify the full switches for Rex's command line might get a little tedious after a while; as a result, there are a number of ways of shortening Rex's command line.

Abbreviations

Each of the command line options can be abbreviated; you only need specify enough of the command line option to identify it. For instance:

```
rex -fetch                → rex -f
rex -fetch binkp ftp      → rex -f b f
rex -fetch 1:342/806      → rex -f 1:342/806 -s 1:342/806
    -send 1:342/806
rex -fetch mail -trim     → rex -f m -t
rexw +autodial            → rexw +autod
```

Merging

The fetch and send command line options can also be merged when they're being applied to the same set of nodes.

```
rex -f m -s m            → rex -fs m
rex -f f 1:342/806 -s f  → rex -fs f -f 1:342/806
```

Running Rex

Interacting with Rex

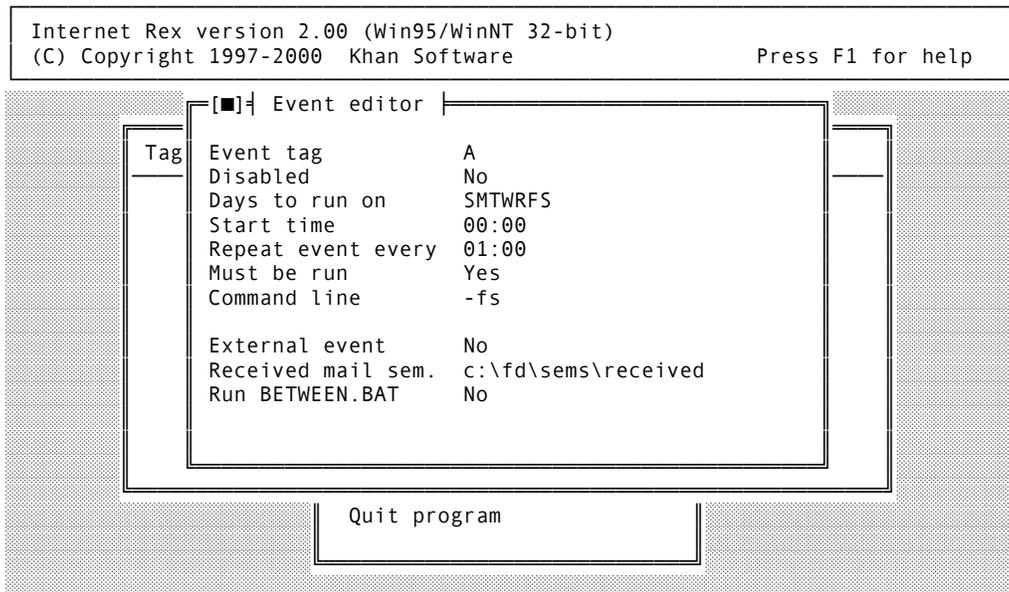
While Rex is fetching or sending your mail, you can skip or completely abort transfers for a particular node. The following key sequences are available to you:

Alt-S (Ctrl-S for Linux users) Skips the current file being transferred. If the file is being downloaded, it's left wherever it's being picked up from. If the file is being uploaded, it's skipped and sent later on.

Alt-T (Ctrl-T for Linux users) Terminates the current transfer. This not only stops the current file transfer, it drops the connection with whatever server is being used for this node. For FTP, BinkP and SMTP connections, the connection with the server is dropped and the file or message is aborted. For POP3 connections, the download is stopped and the connection is dropped: this has the additional result that all messages previously received from that session will be re-downloaded the next time Rex connects.

Daemon mode

What Rex does in daemon mode depends entirely on what events you've defined for it. Events tell Rex to run with a particular command line or to run a particular program at some predefined time. You configure the events you want run in the *Event manager* part of the configuration program. When you first enter the *Event manager* window, the screen has no entries. To create a new event, hit the **insert** key.



The event menu has two sections: the top describes the event and when it will run. The bottom describes how Rex will react when it receives mail during the event.

Each event has an *Event tag*. This is a single upper or lower case letter or number that is used to select additional routing commands specific to this event. (See the section on

Running Rex

Routing for more details.) Each event also has a day and time at which it's run, and a repeat time. Select the days you want the event run in the *Days to run on* field and the time in the *Start time* field. The start time is the time the event is first run on each of the days it's allowed to run on (in 24-hour clock mode). In *Repeat event every* you give the time between each running of the event. For instance, to have something run every hour, you would give a start time of 0:00 (midnight) and a repeat event every 60 minutes (translated to 1:00). If you only want the event run once a day, just give a repeat event every time of 0:00.

Events can be disabled by setting the *Disabled* field to "yes". Events can also be set to be forced events (*Must be run*). If Rex misses a forced event (because it had been shut down at the time, or the computer was frozen), it will run it immediately the next time it's run in daemon mode. Otherwise, Rex will skip any missed events and run them when they're next scheduled to be run.

What Rex does for any event is governed by the *Command line* field. Here you give the command line you'd have Rex run with if you were running it from a prompt. For instance, to have Rex fetch all mail during a particular event, set the command line to `-f`.

If Rex does receive mail during an event, you can set how it will behave in the bottom part of the menu. Values you set here override any you might have set for the matching fields in the *Files, paths and programs* → *Internet Rex files and paths* menu. The *Received mail semaphore* field gives the name of a file that Rex will create if it receives mail during the event. *Run BETWEEN.BAT* determines whether Rex will run the BETWEEN batch file to process inbound mail if it receives any. *Always run* will tell Rex to run the batch file regardless of whether or not it receives mail during the event.

You can also configure *External events*: these can be used to run another program at the predefined time. In external events, the *Command line* you give should be the same as what you would type at a command prompt to run your program. In multitasking operating systems, you can set whether Rex should wait for the event to run to completion or not using the *Wait for completion* field. (Under DOS, this has no effect: the called program must finish before Rex will resume running in daemon mode.)

Interacting with Rex in daemon mode

In daemon mode, there are some additional keyboard sequences not available in command line mode.

Alt-X Shuts down Internet Rex and its associated daemons. If you're running the (Ctrl-X for SMTP daemon, you may find this takes a little while to happen if Rex is trying Linux users) to deliver mail at the time.

Alt-E This brings up a menu allowing you to run any event from those you've (Ctrl-E for entered in the event manager. Disabled events are also available. Linux users)

Alt-F Brings up an entry box allowing you to specify nodes to fetch mail from. Give (Ctrl-F for the netmail addresses of the nodes you want, or wildcards if you want to

Running Rex

Linux users) specify many nodes at once.

Alt-S Brings up an entry box allowing you to specify nodes to send mail to. Give the
(Ctrl-S for netmail addresses of the nodes you want, or wildcards if you want to specify
Linux users) many nodes at once.

Alt-Q Brings up a queue manager for Rex, allowing you to see what's in Rex's queue
(Ctrl-Q for currently. Hitting enter on any of the entries will bring up specific details of
Linux users) the files queued for sending.

Alt-R Forces a rescan of your mailer's outbound.
(Ctrl-R for
Linux users)

Alt-D Brings up the daemon manager. This allows you to restart or shutdown any
(Ctrl-D for daemon Rex is configured to run. Hit enter on the daemon of your choice to
Linux users) shut it down if it's currently running, or start it up if it isn't.

F1 Switches you to the overview screen displaying the time to the next event and a
short summary of the activities of the various daemons being run.

F2 Takes you to the FTP overview screen, displaying a list of the current FTP
connections.

F3 Takes you to the BinkP overview screen.

F4 Takes you to the SMTP overview screen.

F5 Takes you to the POP3 overview screen.

F8 (In the main overview screen only.) If you specified a log file for Rex to put
outbound connection histories in, this will display that file, nicely formatted.

F9 (In the main overview, BinkP or FTP screens.) If you specified a history log
file for inbound connections, or for the FTP daemon, a log file for transfers,
hitting F9 will bring up a summary of the inbound connections, nicely
formatted.

F10 Provides a context sensitive help screen anywhere in the program.

The number Switches to the appropriate task. (eg. 1 will switch to task #1.) This lets you
keys (1-9) see what's going on with a particular connection in more detail than you would
get in the overview screens.

BBS ↔ Internet gateways

Starting with version 2.0, Internet Rex offers a fully function BBS / Internet gateway. With this you can provide email access for your users, setup mailing lists, gate BBS echos to email or vice versa, even act as a gateway for an entire net. It's done through the *Gateway setup* menu and partly through the *Email setup* menu where you configured your email addresses.

Configuring the gateway

Internet Rex's gateway is broken into two parts: the first part sets up how the gateway will behave in general, where it's accessible and what it will do for particular types of messages. The second part, setting up the gates themselves, adds details on what mail will be sent where and how. Configuration of the first part is done in the *Gateway setup* menu:

```
Internet Rex version 2.00 (Win95/WinNT 32-bit)
(C) Copyright 1997-2000 Khan Software                               Press F1 for help

[■] Gateway setup
Gateway addresses 1:342/806.101, 111:1200/11.
Gateway username  UUCP
Quoted printable  Use when needed
Break lines at   80 characters
Max. message size No size limit
MIME multi-attaches Yes
Name separator   -
Inbound headers
Outbound headers
Message-ID to MSGID Yes
Kill gated netmail Yes
Crash on gated mail Ignore
Advanced setup

Setup gates
News servers
```

The email addresses you setup in the *Email setup* menu determine what your gates will be accessible as on the internet: the first thing you need to do now is specify how people on the BBS side of things will access your gateways. This is done through *Gateway addresses* and the *Gateway username*. Gateway addresses represent the entry and exit point for your gates in the BBS world. Mail sent to the addresses you enter may be gated out to the internet, and mail gated from the internet will show up as being from one of the addresses you enter. To avoid confusion about whether mail should be going to your BBS or the internet, your gateway addresses should not exist in the BBS world. It's best to make them point addresses of one of your system addresses; e.g. if your netmail address were 1:342/806, a good gateway address might be 1:342/806.999 or 1:342/806.1. The *Gateway username* is used to overcome the limitations of the Fidonet Technology Network (FTN) To: header. The To: header in netmail addresses is limited to 36 characters. Unfortunately, many email addresses are longer than this. To get around this, people can enter "Internet Rex" or "IREX" as the To: header and

BBS ↔ Internet gateways

enter the real address they want their mail sent to on the first line of their message. For example:

```
From: Joe User, 1:100/789
To: IREX
Subject: Hi Tom!
-----
To: TomUser@somewhere.com
```

Hi Tom, here's a quick note for you!

The gateway usernames "Internet Rex" and "IREX" are built into Internet Rex already. You can specify a third gateway user in the *Gateway username* field. Since many programs that use gateways default to a gateway username of "UUCP", this is usually a good choice.

How messages change going through the gateway

Internet style messages and FTN style messages differ considerably in how they're constructed and formatted. Because of this, they have to undergo a few changes when they're taken from one world to another. The next few options determine what changes will be made to messages when they're gated.

Quoted printable is an internet message format that allows for things like upper ASCII characters and soft carriage returns, something built into every FTN message automatically. When Rex is gating a message to the internet, it will automatically examine the message and determine if quoted printable is needed to maintain the message's formatting. If you like, you can force Rex to use quoted printable on every message. The only disadvantage to this is that not all mail readers necessarily support the quoted printable format, so people receiving messages from your gateway may find upper ASCII characters escaped with = signs, and an = at the end of every line. You can also force Rex never to use quoted printable in its messages. This means upper ASCII characters will be sent unaltered through the SMTP server: this may not work for all servers.

Quoted printable or not, line breaks are required in internet messages and almost never used in FTN messages. As a result, messages going out to the internet have to have line breaks inserted. Where those breaks are inserted is determined by the *Break lines at* field. You set this to a certain number of characters and Rex will ensure that any message going out to the internet will have no more than that many characters per line. 80 characters per line is a good default value for this.

One limitation FTN style messages have is message size. Many message readers and mail tossers can't handle messages more than 64K in size, whereas internet messages are allowed to be any size the sender likes. If your mail tosser or reader has this limitation, or some of your downlinks receiving gated mail have it, you should set a *Maximum message size*. Enter the number of bytes to restrict messages to and Rex will split large internet messages into a number of smaller FTN messages, all less than the maximum message size.

FTN messages may have limited size, but you can attach as many files as you like to them. Internet messages don't have this flexibility: depending on the format you choose, you may be limited to one attached file per message. UUencoded files fit this profile: when a

BBS ↔ Internet gateways

gate is using UUencoding, Rex will create an additional message for each attached file. MIME encoded files don't have to have this limitation, but some message readers may not correctly interpret multiple files in one message. If you would like your MIME encoded links to try to put multiple files per message, you can set the *MIME multi-attaches* field to "yes" and see what happens.

Internet messages are also limited in the addresses available to them. An email address can't have things like spaces, slashes, upper ASCII characters and so on in them. When Rex has to change an FTN username to an internet username, it changes these illegal characters to another character, the *Name separator* (so named because the character it will change most often is the space separating first and last names). For instance, if you set the name separator to '_', the username "Charles Cruden@1:342/806" might get translated to "Charles_Crudен@somewhere.com". Be careful when setting the name separator to something that might already be in a person's name, a period for example. In cases like that, "Yves St.Claire" would be translated to "Yves.St.Claire" on the internet, but going from the internet to the BBS it would become "Yves St Claire", which might not reach the intended recipient.

Message headers

The most important part of an internet message, to programs at least, is its headers. These are lines of information at the top of the message which specify who wrote the message, who it's for, the subject, what machines it went through, and so on. Some of these have relevance to FTN messages, some don't. You can control what will happen to internet headers going through your gateway with the *Inbound headers*, *Outbound headers* and *Message-ID to MSGID* fields.

Inbound headers lets you control what will happen to the headers of messages coming from the internet. Each of the entries lets you specify a header field and an action. The default entries are:

CC:	Kludge
From:	Copy
Message-ID:	Kludge
Received:	Kludge
Subject:	Strip
To:	Kludge
<i>any other header</i>	Kludge

The "copy" action copies the message header to the top of the FTN message. In this example, this means that the From: line in an internet message will be put at the top of the Fido message it's gated to. The "kludge" action copies the message header to an FTN kludge line. These are invisible lines at the top of netmail or echomail messages that some readers can display if asked to. This way the information isn't lost, and it isn't cluttering up the screen when people don't want to see it. The last action is to "strip" the header from the message: it isn't copied to the FTN message when the internet message is gated. In this example, the Subject: line is stripped because Internet Rex automatically copies that to the FTN message's subject field, so it's no longer needed.

BBS ↔ Internet gateways

Outbound headers performs the same function for messages being gated out to the internet. Here, the headers being dealt with are any that the user might have put at the top of the FTN message they're trying to have gated. For instance, you would want to let them specify the To: header to say who they're sending messages to. But the From: field should be restricted so you don't have users trying to create fake email. The default entries are:

CC:	Overwrite
From:	Copy
Message-ID:	Overwrite
Subject:	Copy
To:	Overwrite
<i>any other header</i>	Strip

Internet Rex automatically tries to generate a certain number of headers on FTN mail gated out to the internet, including Subject, From, To and Message-ID lines. It then examines the user's headers and compares them to the rules given here. If a header is set to be "stripped", it's removed from the message. If it's set to "overwrite", it's included in the message, overwriting whatever header Rex might have generated automatically. If it's set to "copy", the header is copied to the message, assuming Rex hasn't generated that header already.

The *Message-ID to MSGID* line deals with message identification strings. Each message, FTN or internet, has a unique string to identify it, Message-ID being the internet identification string and MSGID being the FTN one. Theoretically, it should be possible to convert internet Message-IDs to FTN MSGIDs without any problems: practically, there are certain message tossers and readers that won't accept an internet style identification string in a MSGID field. If you find that you or your downlinks are having problems with odd message ID strings, set *Message-ID to MSGID* to "no", and Rex will stop putting the internet message ID in the MSGID field.

Message flags

The last two fields deal with FTN message flags. FTN messages can be specified as Kill/Sent so that they're deleted from the *.MSG area once they're sent. For gated messages, Rex ignores this flag and instead checks the *Kill gated netmail* setting. If it's set to "yes", messages gated to the internet will be deleted once they've been sent. Otherwise, they'll be left in your *.MSG area for you to inspect at your leisure.

FTN messages also have the special flags Crash and Immediate. These indicate the message should be sent as soon as possible by the mailer. If Rex is operating in daemon mode, and you've set the *Rex's behaviour* → *CRA/IMM mail* field to "Send immediately", it can try to send gated mail with the crash or immediate flag set as soon as it sees it, provided you've set the *Crash on gated mail* field to "Honour". If you set this field to "Ignore", the message will simply be sent when the next batch of mail is sent. If you would prefer that only messages from your address be allowed to be sent immediately, you can set the field to "Honour, from here only".

BBS ↔ Internet gateways

Setting up news servers

If you intend to have Rex gate newsgroups to and from your echomail areas, you'll need to setup the news servers Rex will use to read and post messages: that's done in the *News servers* menu off the gateway setup menu.

Setting up a news server is fairly similar to setting up an email address. Like email addresses, each server gets a *News server ID* which will be used to identify it elsewhere in the config program. Each server also needs a *Posting host* entry so Rex can construct email addresses when posting messages. When user Joe Blow posts a message to one of the echomail areas you gate to a newsgroup, his email address will be set to Joe.Blow@*posting host*. The posting host needn't be valid (though it may be required to be by some news servers): it just needs to be there.

How messages are read and posted through the news server is determined by the *Type of server*. Rex connects to NNTP news servers over your internet connection: your ISP will tell you what their NNTP news server's address is if they have one. Rex can also read and post to a Soup/Yarn spool, a UUCP spool or a *.BAG spool. Use the type appropriate to however you process news.

If you choose an NNTP news server, you'll need to specify the server's *Hostname*. This is just the machine name for the news server as provided by your ISP, usually something along the lines of news.myisp.com. This is the only required field for NNTP servers. Some servers require their users to log in to the news server before they're allowed to read or post news. If your server is one of these, enter the *Username* and *Password* Rex should use to log in to the server in the appropriate fields. The last field, *Timeout*, overrides Rex's usual timeout for sockets and sets it to however many seconds you give here. News can occasionally take a long time to process, especially if there are a lot of articles to retrieve. If you find downloads of news are timing out, set the timeout value to something higher.

Soup/Yarn news connections need only one setting. Set the *Queue path* to the directory containing your Soup/Yarn spool. It should contain an AREAS file and a REPLIES file.

[Configuring a UUCP connection](#) is the same for news as it is for email. Follow the link for more details. Quickly, the *Queue path* should contain your .X/.D and .Xqt/.Dat/.Cmd files; the *Site name* and *Host name* fields should be set to your UUCP connection's site and host names; the *UUCP grade* should be set to the grade of file you want your news to be posted as; and finally, the *Spool type* should specify how your UUCP spool is stored.

*.BAG connections read and write to .BAGs of news. The *Wildcard* should be a full path with a wildcard following it specifying the files Rex should consider are inbound news articles. Similarly, the *Outbound path* should be a directory which Rex will write outbound .BAG files to when it has something to post.

Setting up gates

Whereas the gateway setup defines how messages will go between your BBS and the internet, the gates themselves define the links that will be created. A gate can be thought of as something linking an FTN address or echo to an internet email address. Sometimes this link involves just one FTN address and one email address, sometimes many at once. The types of

BBS ↔ Internet gateways

links available to you will be determined by the email addresses that you've setup in the *Email setup* menu. Rex supports 7 types of gates:

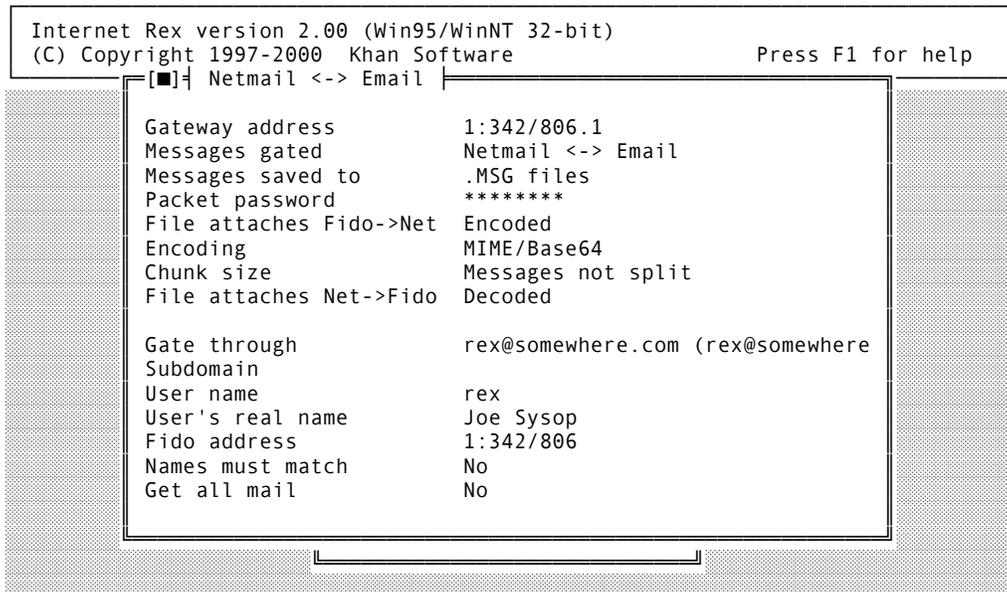
- *Netmail address ↔ Email address.* This links a single netmail address in the FTN world to a single email address. Any message from the netmail address (regardless of the user who sent it) will be converted into email from the email address you select. This sort of gate is useful for sysops who want to use their netmail areas to read and write email.
- *Netmail address ↔ Domain.* This gives each of the users at the netmail address you select an email address in the domain you select. For instance, netmail from Joe.User@1:342/806 might be converted into email from Joe.User@somewhere.com, whereas netmail from Bob.User@1:342/806 would be converted into email from Bob.User@somewhere.com. This sort of gate can be used to give your users access to email. It's only available if you have control over the email addresses at some internet domain.
- *Echomail area ↔ Email address.* This gate takes all the mail sent between two users on the internet and converts it into echomail for the area you specify. This is especially useful for gating mailing lists into echomail areas, and vice versa.
- *Echomail area ↔ Domain.* This gate is similar to the netmail address ↔ domain gate, except instead of taking mail from your netmail folder, it takes mail from a particular echomail conference. This is an alternate way to give your users email access if you don't want to also give them access to writing netmail. As with the netmail address ↔ domain gate, you must have control over the email addresses at some internet domain in order to be able to create one of these gates.
- *Net gate.* A net gate converts netmail from users on multiple BBSs into email messages. The Fido net gate is the best example of this: every user on Fidonet has an email address. If Joe User is a user on 1:342/806.0, then he has the email address Joe.User@p0.f806.n342.z1.fidonet.org. The net gate allows you to build an email address for the users of your net.
- *Mailing list.* This lets you run a mailing list through an email address you provide. The list can also optionally be gated into a local echomail area.
- *Mailing list server.* This type of gate is more of an auto-respond email address. It allows people on the internet to join or leave any mailing lists you're running by sending an email to the server.
- *Newsgroup.* Using the news servers you setup in the main gateway menu, Rex can read messages in a newsgroup and post them to an echomail area, and vice versa.

Although each of the gates serves a different purpose, many of them have common setup options.

Basic gate options

A netmail ↔ email gate is the most common gate chosen to start off with, so lets examine the menu for that gate. Most of the fields are common to all but the mailing list gates.

BBS ↔ Internet gateways



The first thing you must provide is the *Gateway address* for this gate. This is the gate's anchor in the BBS world: netmail or echomail can originate from anywhere, but it must be sent to the gateway address for it to be processed through this gate. You'll be asked to choose from the gateway addresses you entered in the *Gateway setup* menu.

Next you'll have to choose how the gate will operate. Each gate can be set to be bi-directional or unidirectional: that is, mail can either go from the internet to the BBS world only, from the BBS world to the internet only, or in both directions. You might want to do this to prevent people on your BBS from posting to an informational mailing list. By hitting enter on the *Messages gated* field, the arrow sign will change to show which way mail will be allowed to flow.

You must then choose how messages from the internet will be saved on your computer in the *Messages saved to* field. You can either have them written directly to *.MSG files in your netmail area, or saved to a .PKT for your mail tosser to deal with. Several .PKT types are available to you: choose one that will be compatible with your mail tosser. If you choose to save messages to a .PKT, you can specify a *Packet password*. Messages gated from the internet will then be saved to a .PKT with that password in the header. If you give a packet password, Rex will also require that any message going from the BBS world to the internet be in a .PKT with that password in the header.

Finally, you'll have to decide how file attaches will be dealt with through the gate. There are two directions to be concerned about: file attaches from the BBS world and files attached to email messages. Files sent to the gateway from the BBS world can either be encoded and put into email messages, or stripped: you choose which by setting the *File attaches Fido→Net* field. If you choose to allow file attaches, you'll need to specify what encoding method to use and how big attached files can be before they're split into separate messages. The two internet standard *Encodings*, UUencoding and MIME/Base64 encoding are available. Large files can either always be sent in one message, or you can specify a non-zero *Chunk size*, in which case the file will be split into pieces as large as you specify and sent in multiple messages.

BBS ↔ Internet gateways

For email messages passing through the gate into the BBS world, Rex can automatically decode attached files. What happens to those files is determined by the *File attaches Net→Fido* field. They can either be decoded and attached to the BBS message that's created, or left in the message itself. Some gates won't allow this field to be set: gates involving echomail areas, for instance, must leave attached files in the message, as file attaches aren't allowed in echomail areas.

This covers the common setup options for most gates. Now let's examine the options specific to each type of gate.

Netmail ↔ email gates

The basic options presented above have configured most of the details of the netmail ↔ email gate: what's left is to define who the gate is for in the BBS world and the internet world.

The internet world's anchor is set by the *Gate through*, *Subdomain* and *User name* fields. Together, these three fields set the internet address that mail for this gate will be downloaded from and uploaded as. *Gate through* allows you to choose which of your email addresses the gate will work through. If that email address has been set with the *Domain controlled here* flag on you'll have to specify which *User name* and *Subdomain* you'd like to use for the address. The three are combined as follows:

User name@Subdomain.Gate through

So, suppose you set the user name to "Joe.User", the subdomain to "somewhere" and your email address controlled the domain "mydomain.com". Then this gate's address would be "Joe.User@somewhere.mydomain.com". All the email that user received would be downloaded and converted into netmail. "Joe.User@somewhere.mydomain.com" would also be the address used for all the mail that was sent from the BBS world to the internet.

With the internet address set, all that's left is to define which BBS user will be using the gate. This is set in the *Fido address* field. Any netmail from that address to the gateway address will be sent through the gate.

The *User's real name* field will be used in two places. The first is as the real name set in the From: field of the email created by this gate. Following the example above, if the real name were set to "Joe User", the From: field for email messages would read "From: Joe User <Joe.User@somewhere.mydomain.com>". The real name field will also be used in the To: header of the netmail messages generated: "To: Joe User, 1:342/806.0". Also, if you set the *Names must match* field to "yes", only netmail from the user in *User's real name* will be allowed to pass through the gate. You can use this to setup gates for multiple users on the same BBS.

If the email address you're gating through is going to be used exclusively by Internet Rex, you can have Rex download all the mail in the mailbox when it checks for new mail using the *Get all mail* field. Otherwise, Rex will only download mail addressed specifically to your email address: this will leave out things like mailing list mail and certain types of spam email.

BBS ↔ Internet gateways

Netmail ↔ domain gates

Netmail ↔ domain gates function nearly the same as netmail ↔ email gates, except that the username in To: headers is preserved across the gate. For example, suppose you have two pieces of email in your mailbox, one to Joe.User@somewhere.com and one to Jim.User@somewhere.com. A netmail ↔ gate would translate both emails into netmail messages “To: Joe User, 1:342/806.0”. A netmail ↔ domain gate will translate the first email into netmail “To: Joe User”, the second into netmail “To: Jim User”. Because of this, a netmail ↔ domain gate is a great way to give email access to your users, assuming they also have access to netmail. Also as a result of the translation, netmail ↔ domain gates are only available if you have an email address setup with the *Domain controlled here* field set to “yes”.

The internet side of the gate is controlled by the *Gate through* and *Subdomain* fields. These determine what email Rex will select to go through the gate. Selecting *Gate through* will present you with a list of the email addresses you have configured with domain control. Once you’ve select one of these, you can optionally enter a *Subdomain* for the gate which will be inserted between the @ sign and your email address’s domain. For example, if your email address controlled the domain somewhere.com and you set the subdomain to mybbs, then the gate would deal only with messages addressed to someone @mybbs.somewhere.com.

The BBS side of the gate is setup in the *Fido address* field. Here you enter the netmail address messages must come from in order to pass through the gate. For example, if you set the fido address to 1:342/806 and the gateway address to 1:342/806.1, then only netmail messages from 1:342/806 to 1:342/806.1 would pass through the gate. The fido address is also the netmail address gated email will be sent to.

Finally, you can enable or disable subdomains of the domain you’ve defined here with the *Subdomains allowed* field. With this enabled, the gate will also process mail to subdomains of the domain defined. For example, if the gate’s domain were mybbs.somewhere.com as given above, enabling subdomains would mean that email messages to someone @joesbbs.mybbs.somewhere.com would also go through the gate.

Echomail ↔ email gates

Echomail ↔ email gates link messages posted in an echomail area of yours to messages sent between two email addresses. This is different from the behaviour of a netmail ↔ email gate, which will process email from any address: for echomail ↔ email gates, the email can be to any address, but it must be from a particular email address to be processed. This is ideal for gating mailing lists, which generally aren’t addressed to anyone but always come from the same address.

As with previous gates, you first have to tell Rex which email address mail for this gate will be downloaded through in the *Gate through*, *Subdomain* and *Username* fields. The values you enter in these fields will be combined to form an email address as follows:

Username@Subdomain.Gate through

or if no subdomain is given, just

Username@Gate through

BBS ↔ Internet gateways

The subdomain and username fields will only be available if the email address you selected in *Gate through* controls a domain.

On the BBS side of things, you need to configure the *Echomail area* and *Fido address* Rex will require for mail to be gated. To pass mail through the gate, use your mail tosser to export messages from the echomail area into a .PKT and have it send those messages to the gateway address you select for this gate. You then enter the area tag from the BBS in the *Echomail area* field, and the address your mail tosser will use to create the .PKTs in the *Fido address* field.

The last field that needs to be configured is the email address mail Rex will *Send mail to* when mail passes through the gate - this is also the address email for the gate must come from to be gated into the BBS world. For a mailing list, this would just be the list's email address.

The last two fields, *Organisation as origin* and *Signature line*, are used to help format messages that are going into the echomail area. Setting *Organisation as origin* to "yes" tells Rex to use the information in the email's Organisation: header to set the origin line on echomail messages. If it's set to "no", or if Rex can't find an Organisation: header, it will set the origin line to "* Origin: The gateway at *Your system name (Your gateway address)*". The *Signature line* field lets you give a string of characters that Rex will use to identify the start of a signature in the email message. Signatures are blocks of text added to the end of some email messages which give standard information about the mailing list, or pithy remarks, or something similar: most of the time, you don't want these to show up in the echomail area as they aren't really relevant. If you give a string of characters in the signature line and Rex spots a line starting with those characters, it will delete everything in the message from that line onwards.

Echomail ↔ domain gates

An echomail ↔ domain gate works much like a netmail ↔ domain gate, gating not only the message text, but turning the BBS username into an internet username and vice versa. You can use this sort of gate as an alternative way of giving email access to your users if you don't want to give them write access to your netmail area. Unlike regular echomail, all the messages created on the BBS side will have the private flag set so that only the intended recipient of the message can read it. Note that, like the netmail ↔ domain gate, this type of gate will only be available if you have an email address with the *Domain controlled here* flag set to "yes".

The first thing to setup is the internet domain that will be used for this gate. This is set in the *Gate through* and *Subdomain* fields. Mail addressed to someone @*Subdomain.Gate through* will pass through the gate. (If you leave the subdomain blank, the gate will simply take mail to someone @*Gate through*.)

On the BBS side of things, you must give the *Echomail area* that the gate will use to read and write messages and the *Fido address* mail must originate from to pass through the gate.

Finally, the *Organisation as origin* field allows you to use the Organisation: header in email messages passing through the gate to set the origin line of the echomail messages that are generated. Set to "yes", an email message with the header "Organisation: My company's

BBS ↔ Internet gateways

name” would show up with an origin line of “ * Origin: My company’s name”. Otherwise, if no Organisation: header is available or the field is set to “no”, the origin line will read “ * Origin: The gateway at *Your system name (Your gateway address)*”.

Net gates

Net gates are a special type of gate which provide an email address to every user in a particular set of netmail addresses. Fidonet offers this service for many of its nets. For the netmail address *zone:net/node.point*, users of that BBS have the email address “*User’s name@ppoint.fnode.nnet.zzone.fidonet.org*”. Assuming you have control of an internet domain, you can setup the same sort of gate for users of your BBS network. This sort of gate will only be available if you have an email address configured with the *Domain controlled here* flag set to “yes”.

Net gates don’t require much information to setup, as most of it is built into the netmail and email addresses that go through it. What is required is the internet domain that the gate will be based in: for the Fido example given above, this is the “fidonet.org” at the end of the address. The domain for your gate will be determined by the *Gate through* and *Subdomain* fields. For example, if you gated through an email address controlling the domain somewhere.com and added the subdomain mynet, users of your net gate would have addresses that looked something like “*Username@ppoint.fnode.nnet.zzone.mynet.somewhere.com*”.

The other field for this gate is *Addresses gated*. This lets you specify which netmail addresses will be allowed to use the net gate. You can limit this to just one BBS, or specify a number of BBSs with a wildcard. For example, to define a net gate for net 111, you would set *Addresses gated* to “111:*”. If you just wanted to setup a gate for the nodes in your local net, you might set it to something like “1:342/*”.

Mailing lists

A mailing list gate both is and isn’t a real gate. It isn’t a real gate in that its main function is to be a mailing list controlled by you: the *Mailing list setup* field for this gate demonstrates that. Selecting that will bring up a menu allowing you to configure all sorts of details about your mailing list: who receives and who can send messages, where it’s archived, whether people must renew subscriptions to it and so on. The details of setting up a mailing list are covered later on in this manual in the SMTP server section on [Configuring a mailing list](#): please take a look there for more details on how to setup the mailing list part of this gate.

A mailing list gate is a real gate in that not only do you control the mailing list, you can optionally have it copied into an echomail area on your system, and have messages from the echomail area posted to the list. In this respect, a lot of the configuration for a mailing list gate is similar to that of the echomail ↔ email gate.

For each mailing list you should define the *Mailing list’s name*: this provides a comprehensible name for the mailing list when information about it is required. For instance, a mailing list for Internet Rex might have the address *irex@xanadu.v-wave.com*, which wouldn’t tell people much: to call it the “Internet Rex mailing list” says much more. You can

BBS ↔ Internet gateways

also optionally give mailing lists an *Echomail area* to be gated into. If you choose to do this, you'll also need to provide the *Fido address* to expect mail from and write mail to.

The mailing list's address is set by the *Gate through*, *Subdomain* and *Username* fields. When configured, they're combined as follows to form the mailing list's email address:

Username@Subdomain.Gate through

If the subdomain field is blank, it becomes simply

Username@Gate through

The subdomain and username fields will only be configurable if the email address you chose in *Gate through* had domain control.

The final configurable field in the mailing list setup menu is *Organisation as origin*, only available if you've also set the mailing list to be gated into an echomail area. In this case, setting *Organisation as origin* to "yes" will direct Rex to search gated emails for an Organisation: header and use that as the origin line. If no Organisation: header is found, or the field is set to no, the origin line for gated emails will appear as "* Origin: The gateway at *Your system name (Your gateway address)*".

Mailing list server

A mailing list server "gate" isn't a gate at all, but a service you can provide on a particular email address. Mailing list servers read email messages and subscribe or unsubscribe people to mailing lists: if you have any mailing list gates setup, you can use a mailing list server "gate" to allow people to join your list remotely by just sending some email. List servers also provide lists of the mailing lists you're running and help to those who request it.

The main thing required for a mailing list server is an email address to process mail at. You configure this in the *Gate through*, *Subdomain* and *Username* fields. The values you enter for these fields will be put together to form an email address as follows:

Username@Subdomain.Gate through

If the subdomain field is blank, it becomes simply

Username@Gate through

The subdomain and username fields will only be configurable if the email address you chose in *Gate through* had domain control.

The last field for mailing list servers, *Help on empty messages*, determines what the list server will do if it receives an email message that doesn't contain any commands it recognises. (List server commands are lines like "JOIN mailinglist" or "LEAVE mailinglist" in the body or subject line of the email message.) With this field set to no, the list server will simply do nothing but quote the user's message back to them. With the field set to yes, a help message will be added to the end of the message detailing what commands the server understands. Normally this help message is generated by Rex, but you can override the contents of the message by putting them in a file called `listserv.txt` and putting that file in Rex's home directory.

BBS ↔ Internet gateways

Newsgroups

Newsgroup gates are just that: a gateway between a USEnet newsgroup and a BBS echomail area. This allows your users to read any newsgroup you select and to have their posts to the echomail area show up in the newsgroup for others to respond to.

The first thing you'll need is to decide which group to gate and where to post it. The group is entered in the *News group* field and the area in the *Echomail area* field. You'll also need to tell Rex how to read and post messages by telling it which *News server* to use.

If you chose an NNTP based news server, the *Watermark* field will be available to you. This keeps track of the most recent message on the news server that's been gated. When you first setup a newsgroup gate, this will be set to 0. With this setup, Rex will retrieve every message in the newsgroup the first time it logs into the server. For some groups, this could represent thousands of messages. If you'd prefer Rex to start at the most recent message, set this to a high value (in the millions), or if you'd like just the last *n* messages downloaded, set it to *-n*. For example, to have the last 50 messages downloaded, set the watermark to -50.

The *Organization as origin* and *Origin* fields let you control what the origin line of the messages posted to the echomail area will look like. By default (with *Organization as origin* set to "no" and *Origin* left blank), the origin line will be "The gateway at *your BBS name (your BBS address)*". You can pick your own default origin by setting the *Origin* line. Setting *Organization as origin* tells Rex to override whatever the default is and use the *Organization:* header of the posted message to set the origin. If there's no *Organization:* header, the default is used.

Some newsgroups are moderated. This means users can't post to the group directly: their posts must be emailed to the moderator, who then posts them. If you want to allow your users to post to a moderated group, you'll need to set the *Associated gate* and *Moderator* fields. The *Moderator* field is simply the email address of the moderator. The *Associated gate* field controls what address will be used in the *From:* field of the messages sent to the moderator. This field must contain a valid email address, so to tell Rex how to build a valid address, you give another of the gates you've configured here. Rex will then treat the message as if it were being gated to email through that associated gate. So if you associated it with an echomail ↔ domain gate, the *From:* field would be set to *Joe.User@whatever domain the gate uses*. If you associated it with a netmail ↔ email gate, it would be set to whatever email address had been configured in that gate.

The *Associated gate* field can also be used in unmoderated groups to override the default *From:* field Rex builds for newsgroups (namely *Joe.User@news server's posting host*).

Some sample gates

Gateways are, to say the least, a bit complicated, and as such it's not always immediately obvious how to go about doing what you want to do. For those just starting out with gateways, here are a few commonly encountered gateways and a step by step guide to configuring them.

BBS ↔ Internet gateways

The personal email to netmail gateway

This gateway is used to let you read and write your email through your netmail area.

To start out with, you should have setup your email address in the *Email setup* menu. Let's suppose your netmail address is 1:2/3.

1. Go to the *Gateway setup* menu, and add 1:2/3.1 to the *Gateway addresses* list. If you have a point with that address, pick a point address that isn't already taken.
2. Leave the rest of the *Gateway setup* menu at its default settings. If you've already changed things and aren't sure how to change things back, you can reset the menu by quitting the configuration program, deleting `gateway.rex` and restarting it. This will also delete any gates you've setup, so only use it as a last resort.
3. Select *Setup gates*, hit **insert** and choose "Netmail address ↔ Email address".
4. Most of the fields should already have been configured for you. There are a few things that may need changing. If your mailer or BBS program doesn't support a *.MSG style netmail area, change the *Messages saved to* field to "Type 2 packet".
5. If you have setup more than one email address in *Email setup*, you'll need to select the one you want to gate for the *Gate through* field.
6. Enter your name in the *User's real name* field.
7. In the example here, our netmail address is 1:2/3, so that gets put in the *Fido address* field.
8. Finally, it's a good idea to set *Get all mail* to "yes", to make sure all the messages sent to your email address are actually downloaded and converted to netmail.

The gate is now configured. Email Rex doesn't identify as being for itself will be downloaded and converted into netmail. To send a message, create a new netmail message, put the email address of the person you're writing to in the To: header of the message, and send the netmail to your gateway address (1:2/3.1 in this case).

Gating a mailing list to and from an echomail area

Another common use for gates is linking an internet mailing list to a BBS echomail area. Messages posted to the mailing list will also be posted to the echomail area, and messages written by users in the echomail area will be converted into postings to the mailing list. Because this gate is done through a single email address, the messages will be posted as being from your email address, but the name associated with the email address will be that of the user from the echomail area.

Let's assume as before that your netmail address is 1:2/3 and you've setup an email address to post things through in the *Email setup* menu. Let's now choose to gate the echomail area AREA to the internet mailing list whose address is list@somewhere.com.

1. If you haven't already configured a gateway address, add 1:2/3.1 to the *Gateway addresses* list in *Gateway setup*. If you have a point with that address, choose a different gateway address.
2. Select *Setup gates*, hit **insert** and choose "Echomail area ↔ Email address".
3. Most of the fields should have been configured for you. If you entered multiple gateway addresses, choose the one you want to use for this gate in the *Gateway address* field. Remember it.

BBS ↔ Internet gateways

4. If you configured multiple email addresses, select the one that will be receiving messages from the mailing list for the *Gate through* field.
5. The *Echomail area* we want is AREA.
6. Our *Fido address* is 1:2/3.
7. The gateway will be *Sending mail to* (and receiving mail from) lists@somewhere.com.
8. That configures all that's necessary in the gate. Your mail tosser must still be configured. For the echomail area AREA, add the gateway address from step 3 (1:2/3.1) as a downlink and configure it to create raw .PKTs for that link. Arcmail bundles will **not** work.

Your mailing list is now gated. When messages for it are received, a .PKT file will be built: have your mail tosser toss it and the messages will appear in the echomail area. If you want to get fancy, you can try setting *Organisation as origin* to "yes" and see what happens. If your mailing list includes a signature (a piece of text added to the end of every message), you can try to strip that out of the messages gated to the echomail area by setting the *Signature ID* field to the first line of the signature.

Giving your users email addresses

This gate requires that you have control of an internet domain: if you just have the one email address available to you, you can't very well have **all** your BBS users (and you!) using that one address. If you have control of an internet domain, this gate will let you set an echomail area where users can read and write email messages.

Let's assume you have control of the internet domain mybbs.com and that your netmail address is 1:2/3. Your users will be reading and writing email in the echomail area EMAIL.

1. The first thing you need to do is setup an email address in *Email setup* that will receive all the mail for the mybbs.com domain. If you're using Rex as your SMTP server, a good way to do this is [Configuring a domain store and forward user](#) who controls the mybbs.com domain and using that user in *Email setup*. If you choose to do it this way, the *Recipient* line should read "Copy from X-Rcpt:".
2. If you haven't already configured a gateway address, add 1:2/3.1 to the *Gateway addresses* list of the *Gateway setup* menu. If you have a point with that address, pick one that doesn't conflict.
3. Choose *Setup gates*, hit **insert** and choose "Echomail area ↔ Domain".
4. Most of the fields will already have been filled in for you. If you have multiple gateway addresses setup, choose the one you want to use for this gate and remember it.
5. If it hasn't already been selected, go to the *Gate through* field and choose the email address you setup to receive mail for your domain.
6. The *Fido address* is 1:2/3.
7. The *Echomail area* is EMAIL.
8. The gate itself is now configured. You still have to configure your mail tosser to work correctly with the gate. Take the gateway address from step 4 and add it to your mail tosser as a new downlink.
9. Add the echomail area EMAIL to your tosser. Set your address as 1:2/3 and select the gateway address (1:2/3.1) as a downlink.

Your users now have email addresses @mybbs.com. (For instance, Joe User on your BBS has the email address Joe.User@mybbs.com.) They can write messages to people on the internet

BBS ↔ Internet gateways

by writing a message in EMAIL To: the email address they want. To have the message sent, run your mail tosser and Rex will pick up the packet and process it. Inbound messages will be written to a .PKT that your mail tosser can toss to the EMAIL area.

Giving email addresses to a whole network

This sort of gate lets you give an email address to everyone in a whole BBS style network. Anyone who can write netmail can has an email address. As with the gate for giving email addresses to your users, you'll have to have control of an internet domain to do this.

Let's assume that you control the internet domain mybbs.com and that your netmail address for the network you're gating is 12:13/14.

1. You first need to setup an email address in *Email setup* that will be receiving the mail for your domain. If you're using Rex's SMTP server to control your domain, a good way to do this is [Configuring a domain store and forward user](#) who controls your domain, and using that email address in *Email setup*. If you choose to do it that way, set the *Recipient* line to "Copy from X-Rcpt:".
2. Next, you need to choose a gateway address for your net: 12:13/14.1 would be good. (Pick a different one if that one's taken.) Add that address to the *Gateway addresses* list in the *Gateway setup* menu.
3. Choose *Setup gates*, hit **insert** and add a "Net gate".
4. Set the *Gateway address* to the one you chose in step 2.
5. Set *Gate through* to the email address you setup in step 1.
6. Set the *Addresses gated* to a wildcard matching the addresses you want to allow to use your gate. In this case, 12:* would be good, allowing anyone in the net to use the gate.

You're done! Now when someone in your net wants to write email, they write a netmail message To: the email address they want, set the netmail address of the recipient to your gateway address (12:13/14.1) and send the message. If they were at, say, 12:10/10, their netmail would be converted into email from Joe.User@f10.n10.z12.mybbs.com. The same works in reverse: email you received to, say, Bob.User@f13.n9.z12.mybbs.com would be converted into netmail for Bob User, 12:9/13.

Getting to know Rex

The preceding sections should have given you all the information you need to know to get a quick connection setup and working in Rex. However, that's hardly all there is to Rex. Encryption, file requests, reliable connections, netmail gating: all these things are available to you with Internet Rex. Now that the initial setting up has been done, we can delve a bit more into how to get these features running for you and what you can do with them.

Some features of Rex are only accessible when the *User level* setting is a certain level or higher. For features like these, the user level required to access them will be indicated in brackets beside them.

Routing

`Route.rex`

The node manager only allows you to define specific nodes to whom you want Rex to send mail. While this will cover most people's requirements, you may find you want to route mail for people beside yourself to nodes in your node manager – for instance, if you were the hub for a network and you want to provide routed netmail for your downlinks. To accomplish this, you need to setup some routing statements in Rex's routing file.

All the routing in Internet Rex is controlled through the file `route.rex` located in Rex's home directory. This is a simple text file you can modify with any text editor you have handy. In it you define the routing structure you want for Rex by default (global routing setup) and the way you want it to route mail for particular events (event specific commands).

There are three basic routing commands: `route-to`, `hold` and `unhold`. Each of these can be followed by a modifying `except` command. Their use is similar to the routing commands of most mailers.

route-to

The syntax for the `route-to` command is:

```
route-to <destination> <address(es)> [<address(es)>...]
[except <address(es)> [<address(es)>...]]
```

Destination is the address you would like mail sent to. This must be an address belonging to one of the nodes in the node manager. *Address(es)* represents the addresses whose mail should be sent to the destination node. You can specify a single netmail address, or use wildcards to specify a range of possibilities. Individual netmail addresses can be abbreviated: if no net or zone is specified, they will be taken from the previous address. Any number of wildcards can be added after the `route-to` command. You can also add an `except` statement after the `route-to` command. Any addresses you add after the word `except` will not be routed to the destination address, even if they match one of the addresses following the `route-to` command. An example:

```
route-to 1:17/0 1:342/* 1:153/* 1:140/1 2 4 14
except 1:342/8??
```

Getting to know Rex

This command routes mail from any addresses in net 342 or 153, and the addresses 1:140/1, 1:140/2, 1:140/4 and 1:140/14 to the node 1:17/0 in the node manager. There are some exceptions though: if mail is from a node in the 1:342/800 hub area, it is left alone.

hold

The syntax for the hold command is:

```
hold <address(es)> [<address(es)>...]
    [except <address(es)> [<address(es)>...]]
```

The *addresses* work the same as in the route-to command. They can be wildcards or single addresses – addresses matching the except list will not be included. The hold command directs Rex not to send mail for nodes matching the addresses given, even if they're in the node manager. Usually you won't use a hold command in the global command section: it's best kept for particular events.

The hold command also has two subcommands: `hold-mail` and `hold-files`. These don't take additional addresses: they are simply one-line commands. Hold-mail directs Rex to hold all mail bundles, messages and packets waiting to be sent. Hold-files does the same only for files: any file which isn't a mail bundle or packet will be held. These are useful for people who are hubbing for networks who don't want to send large files during particular times of the day, but want mail and echomail to move quickly.

unhold

The syntax for unhold is the same as for hold.

Unhold is around to reverse the effect of a hold command. Mail previously marked hold can be unmarked by adding an unhold command for the addresses you'd like sent.

As with hold, unhold has two subcommands: `unhold-mail` and `unhold-files`. As might be expected, these commands reverse the effect of the hold-mail and hold-files commands.

route.rex structure

`route.rex` is a text file divided into sections. The first set of routing commands are the global commands, modifying Rex's behaviour when it isn't running a particular event and setting down the basic behaviour events sections modify. After the global commands come routing commands for each event. These are separated by `event` commands followed by an event tag. The event tag defines which event the commands following it will apply to.

Comments can be entered in the file by starting a line with a semicolon (;). Here is a sample routing file.

```
; Routing file for 1:9999/123

route-to 1:456/0 1:* 2:* 3:* 4:* 5:* 6:*
    except 2:250/25 3:640/123
    except 4:235/*
hold 2:250/25

event A
unhold 2:250/25
```

Getting to know Rex

```
event B
hold 3:640/123
```

The file begins with a comment describing what it is. It then sets up routing for the node 1:456/0 in the node manager. Any mail 1:9999/123 creates for someone in Fidonet will be routed to 1:456/0, with one or two exceptions. Mail for 2:250/25, 3:640/123 and all members of net 4:235 won't be sent to 1:456/0. Presumably these are other nodes in the node manager, and Rex will send their mail directly to them. Mail for 2:250/25 is put on hold by default. If Rex were run with the `-s` command line option and mail for 2:250/25 was waiting, it wouldn't be sent.

The next set of commands defines how Rex will behave during the event with the tag "A". During this event, mail for 2:250/25 is unheld: that means that if mail is waiting for 2:250/25 and the event sends mail, 2:250/25 will get its mail. What this does is set things up so that mail for 2:250/25 will only be sent during event A.

The last set of commands defines behaviour for Rex during event "B". During this event, mail for 3:640/123 is left on hold: even if there's mail for them, it won't be sent. This does the opposite of what the commands for event A do: mail for 3:640/123 will be sent by default, except if event B is running.

Customising Rex's appearance

Rex's behaviour → *Rexcfg's appearance*

Rex's behaviour → *Rex's appearance*

So you start up Rex's configuration program, or Rex itself, and go "Yuck! My God, what a terrible colour scheme! Blind monkeys with crayons could do better than that!" Well, hopefully you didn't, but just in case you did, or in case you feel like trying your hand at a colour scheme for Rex, you're in luck. Internet Rex has customisable colour schemes: just about everything about the program can have its colour changed. Opening these menus, you'll be presented with the opportunity to choose from a number of different built-in colour schemes, create your own, save your creations to disk or load in someone else's creation to try.

The first choice in each menu, *Use upper ASCII*, is more of interest to people using Rex on a terminal that doesn't support the usual IBM upper ASCII character set. (Linux terminals are a good example of this: there's some special kludging that can be done to get IBM characters, but most of the time it uses a UNIX character set.) For these people, turning off upper ASCII will make Rex replace its line drawing and graphics characters with lower ASCII characters that should display the same on all terminals. Unlike the other options on this screen, this feature applies to Rex itself as well.

The *Colour scheme* field lets you choose which colour scheme the configuration program will use. You can cycle through a number of different built-in schemes here to choose the one you like: the change won't be applied to the program until after you shut it down and restart. The last choice in the list of colour schemes is "Custom". When this choice is made, the remaining fields on the screen become active and you can choose each and assign a colour or character to it.

Getting to know Rex

When building your own schemes you can either start from scratch, copy in one of the built-in schemes (select the scheme you want to start from in the *Colour scheme* field, then choose the *Copy current scheme to custom* field and hit return), or load a colour scheme from an external file. If you want to do the latter, set the *Colour scheme* to “Custom”, then choose *Save/load current scheme to/from file* and select *Load colour scheme from file*. This will present you with a list of the Colour Scheme Files (*.CSF) in the current directory: choose one to copy to the current scheme by hitting **enter**, or quit out by hitting **escape**. When you’re done creating your masterpiece you can save it to a file from the same menu: choose *Save colour scheme to file* instead and give a filename (Rex will add .CSF if it isn’t in the name) and Rex will save it.

Multiuser email addresses

Email setup → (address) → *Domain controlled here*

Email setup → (address) → *Multiuser address*

These days, everyone and their dog has an email address. Many people have several, and some have all these addresses forwarded to a single mailbox. Some people also have mailboxes that receive mail for an entire domain. If you’re one of these people, you probably have a multiuser mailbox, where you receive mail for more than one address through a single mailbox. To deal with situations like this, Rex can separate out mail for one particular address from all the rest to make sure it only looks at the mail you want it to. If you control an internet domain through a single mailbox, you can also configure Rex to take advantage of that. To do this, you need to deal with the *Domain controlled here* and *Multiuser address* fields in the *Email setup* menu.

First, Rex needs to know whether you control a domain through your mailbox or you just receive mail for multiple addresses. Controlling an internet domain means you are the final authority on who gets email addresses at a particular domain. For instance, if you controlled the domain “somewhere.com”, anyone with an email address ending in @somewhere.com got their email address from you. If you control an internet domain, find the email address you setup for that domain and set the *Domain controlled here* field to “yes”. If you simply receive mail for multiple addresses through a single mailbox, leave *Domain controlled here* at “no” and set *Multiuser address* to “yes”.

When you do this, a third field, *Recipient*, may appear. Rex uses the information you enter here to determine exactly who a piece of email is for. This is necessary because a lot of email (mailing list postings, some spam, group aliases and so on) show up in your mailbox without a To: header on the message, or with a To: header which doesn’t accurately reflect who the email is for. You first need to figure out where in the message the real recipient of the email is located. Some mail hosts put in a special header which contains this information (X-Orcpt: or X-Rcpt:), some put it in as part of another header (in the Received: header for instance). If your mail host puts the information in a header field by itself, hit enter on the *Recipient* field, and select “Copy from” from the list box which appears, then enter the header field the recipient information is in. If the recipient information is just part of the information in a particular field, choose “Search” instead, then enter the field to look in and a string which

Getting to know Rex

will identify the start of the recipient email address. For instance, some mail hosts put the recipient in the Received: header field, something like this:

```
Received: from mail.somewhere.com (1.2.3.4) for user@mydomain.somewhere.com
```

In this case, the recipient is user@mydomain.somewhere.com and it follows the word “for” in the Received: header. So the information for the *Recipient* field might look like

```
Search Received: for for
```

with the second for having a space before and after it. The spaces tell Rex to look for the word for all by itself instead of accepting it as part of another word. Rex will then take the first email address it finds after that word as being the real recipient of the email message.

Remapping drives and directories

Files, paths and programs → *Drive mappings*

Just because one operating system or one computer sees your hard drive setup one way doesn't mean they'll all see them setup the same way. Internet Rex includes some facilities for mapping drives at runtime so that what you refer to as drive E: on one machine and drive N: on another can both be accessed by Rex. These mappings can then be chosen at runtime using the `-map` command line option.

There are two categories of mapping Rex does: drive to drive mappings and drive to directory mappings. The first is intended to be used on systems which boot multiple operating systems or where Rex is used on multiple machines: it maps all occurrences of one drive letter to another. The second is intended to be used on operating systems which don't have drive letters, like Linux: this maps all occurrences of drive letters to a directory somewhere on the hard drive.

Mapping drives to drives (DOS, OS/2, Windows)

To create a new drive mapping, select *Edit mappings*: if you have a drive map defined already, hit the **insert** key to bring up the *Drive maps* menu. At the top of this menu is the *Map label* field where you can give the drive map an appropriate name. This label will be used elsewhere to refer to this particular drive map. All you then have to do is choose *Edit mapping* and you'll go into the drive map list: enter the drive you want to remap and the drive to map it to. You can do this as many times as you like for as many drives as you like. From then on, if Rex sees a reference to a file on a drive in the first column, it will look for that file in the same directory on the drive listed in the second column. For instance, mapping drive Q to drive E means that if Rex saw a reference to the file Q:\BBS\FILES\TEST.ZIP, it would look for the file as E:\BBS\FILES\TEST.ZIP. This applies to **all** drives Rex sees or uses: not just to the paths you enter in Rex's configuration program, but to any files that are in your outbound queue, in your file request list or elsewhere.

The *Drive mappings* menu also has a *Default map* entry. This is the drive mapping Rex will use if none is specified on the command line and the mapping the configuration program will use when checking files or directories to see if they exist.

Getting to know Rex

Mapping drives to directories (Linux)

Setup for the drive to directory mapping list is a little easier. Selecting *Map drives to directories* immediately brings you up in the drive to directory editor. Here you simply enter the drive you want to map and the directory you want to map it to. For instance, if you had your D: drive mounted on /usr/mount/DOSD, you would enter the mapping of D to /usr/mount/DOSD. From then on, if Rex sees a reference to drive D, it will start looking for the file in /usr/mount/DOSD. For instance, the file D:\BBS\FILES\TEST.ZIP would be looked for in /usr/mount/DOSD/FILES/TEST.ZIP.

File requests and email file requests

Files, paths and programs → *File requests*

One of Rex's many useful features is a built in file request processor. This lets any of your connections request files from you through Internet Rex. Rex also includes an email file request processor: this reads certain special types of email and treats them as file requests, sending the files back to the person who sent the email as file attach email messages. All of the information about how Rex should process file requests is included in the *Files, paths and programs* → *File requests* menu. Note that in order for Rex to process any file requests at all, you must first enable the file request processor in the *Rex's behaviour* → *Honour FREQS* field.

A file request comes in the form of a request for a particular file or magic name. For instance, a person might request "BRE0988.ZIP" to get that file, or they might request the magic name "LATEST-BRE" which you could setup to point to the same file. Where Rex looks for files and how it maps magic names is determined by the entries you put in *alias* and *directory list* files.

An alias file is just a text file you can create with whatever editor you like. Here's a listing of a small alias file:

```
LATEST-BRE e:\doors\bre0988.zip
BRE e:\doors\bre0988.zip
FILES d:\bbs\allfiles.zip
SECRET f:\mystuff\secret.zip !carrot
```

On each line you put two things: the alias you want a file to be requestable by and the full name and path of the file the alias refers to. For instance in this alias file, requesting "FILES" would result in the user being sent the file d:\bbs\allfiles.zip.

The last entry in the alias file given is a special one: it has a password attached to the file, as indicated by the last word on the line, !carrot, starting with an exclamation mark. When you add a third entry on a line and start it with an exclamation mark that means that a request for that file must be accompanied by the password, or the file won't be sent back. In this case, a person would have to send a request for "SECRET !CARROT" to be sent back the file f:\mystuff\secret.zip: without the !CARROT part, the request would fail because the password was missing. The password is compared without regard for upper or lower case letters: that's why !CARROT works when the password given was carrot, lower case.

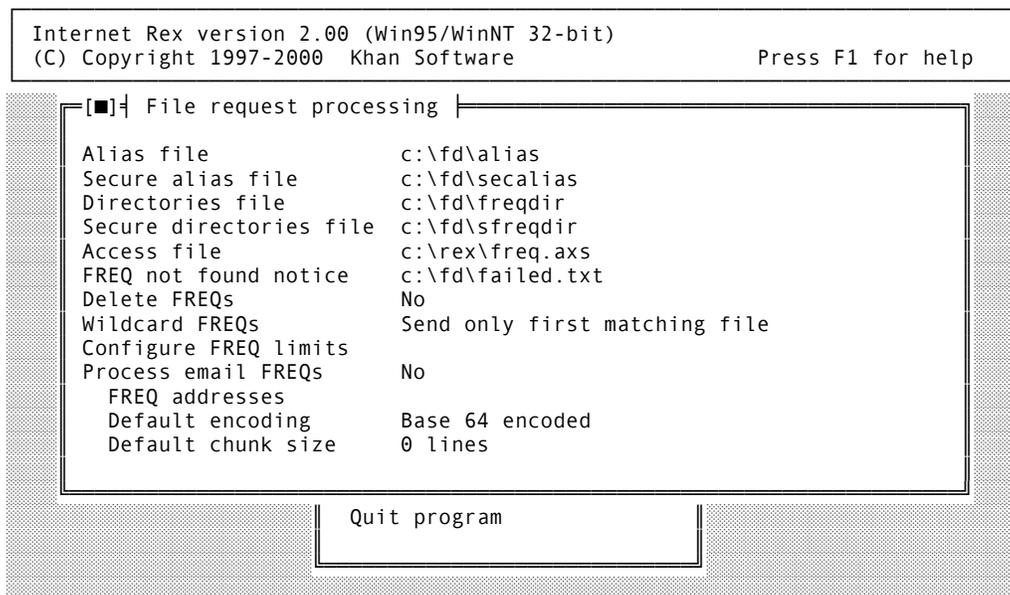
Getting to know Rex

A directory list file is like an alias file: a text file with entries one per line. In this case, the entries are just a list of directories that Rex can search. Here's a sample directory list file:

```
e:\doors
f:\pictures
g:\secret !carrot
```

If a file request doesn't match one of the aliases in the alias file, Rex goes through all the files in all the directories in the directory list file and looks to see if the name matches the file requested. If it does, it gets sent off. Note that the directory list can also have passworded entries like the alias file: if the only instance of a requested file is in the g:\secret directory, the request must include the password carrot to have the file sent back.

That covers how Rex's file request processor works. Now, how do you actually set it all up? Well, let's say you've already setup your alias and directory files. Now you head on into the file request setup menu in *Files, paths and programs* → *File request setup* and are presented with this screen:



You enter the full path and filename of those files in the *Alias file* and *Directories file* fields. There are also two entries for *Secure alias file* and *Secure directories file*: these are used with connections that are deemed secure by Rex (the flag in *Node manager* → *(node)* → *Connection information* → *Secure* is “yes” for the node the request was received from). Rex first searches the alias file, then if the connection is secure, searches the secure alias file; it does the same for the directory file and the secure directory file. If the file being requested isn't found anywhere in the list of directories or aliases you gave Rex, it sends back a message saying that the file request wasn't found. It takes this message from the contents of the file you give in *FREQ not found notice*: this is just a text file that Rex spits out into a message to send back to the user.

On the other hand, let's say the user's request matched a bunch of different files. Rex could either send back the first file it found that matched the request, or send back *all* the files that matched. Which it does is determined by what you set in *Wildcard FREQs*.

Getting to know Rex

Now that Rex has processed the request, there's still the question of what to do with the original file request message. Again, this is configurable: setting *Delete FREQ messages* to "yes" will nuke it, to "no" will leave it in your netmail area for you to deal with at your leisure.

File request limits

Setting up file request processing is all very well and good, but you don't really want users regularly requesting to have your entire hard drive sent back to them over the internet. (Well, maybe you do.... Most people don't though.) You have to set some limits somewhere! Fortunately, Rex lets you do just that in the *Configure FREQ limits* menu and the *Access file*.

In the *Configure FREQ limits* menu you can define limits on how many files and how many kilobytes of files users can request on the basis of each session or each day. The menu divides users into two groups: "known" users, who are listed in the node manager, and "anonymous" users, who aren't. For each of these categories you can specify the maximum number of files they can request and the maximum number of kilobytes of files they can request in the *Maximum requestable kilobytes* and *Maximum requestable files* fields. If you set any of these fields to zero, Rex will interpret that as meaning they can request an unlimited number of files/kilobytes of files. How often these limits are applied is set in the *FREQ limits for known/anonymous nodes applied each* field: you can set this to either "session" or "day". Applying the limit each session means that the maximums are set each time Rex is run. A user couldn't request more than 5 files in one run of Rex, but if you ran Rex 5 times a day, they could request 25 files that day, once in each of the five runs (provided they timed it right). Applying the limit each day means no matter how many times you run Rex in one day, they can only request so many files that day. The next day, the limit is reset and they can request files again.

The *Access file* lets you have a little more fine control over how many files users or classes of users can request. Like the alias and directory list files, the access file is a text file Rex reads when it receives a file request. In this file, each line specifies file request limits for an individual user or class of users. Here's a sample access file:

```
1:342/* 1000 10000 day
9999:*
100:100/0 10 500
100:200/0 10
*@juno.com
```

Each line can have up to four entries on it. The first is the user or set of users to match. Known users will have a netmail address Rex uses to identify them (the one you gave in the *System's address* field of the node manager): Rex will try to match that address against the first field in each line. Note that you can use wildcards in this field, so that if you had a bunch of nodes in net 1:342, the first line in the access file here would apply to all of them. The second entry in the line is the number of files they can request, the third the number of kilobytes, and the fourth should be either "day" or "session", depending on whether you want to apply the limits per day or per session. So for this access file, people in net 1:342 would be able to request 1000 files a day or 10000KB a day.

Getting to know Rex

If some of the entries aren't specified, Rex uses the values you've setup in the *Configure* *FREQ limits* menu to fill them in. So, if you'd specified that known nodes were allowed a maximum of 5 files and 250KB a day, the third and fourth entries would let 100:100/0 request 10 files or 500KB a day, and 100:200/0 10 files or 250KB a day.

By contrast, if nothing except the address to match against is given, Rex assumes that you are denying that user access to file requests. In the above example, users in zone 9999 wouldn't be allowed to request files from you.

Note the entry in the last line gives an email address to match against instead of a netmail address. The access file can be used to apply limits to anonymous users as well as known ones. Since Rex doesn't have a netmail address for these users, it uses the email address the request came from as identification instead. The last entry given here refuses email file requests from anyone whose email address ends in @juno.com.

You don't have to have an access file setup, but you service a lot of file requests, it will probably be useful. Enter the full path and filename of your access file in the *Access file* field of the main file request menu.

Email file requests

One other little field in the file request menu hasn't been discussed: *Process email FREQs*. Email FREQs are email messages in a special format that Rex treats like file requests. If a file is found matching the request, Rex creates an email message to send back to the person who requested the file, with the file attached. The format of the messages is pretty simple: the subject of the message must be **FREQ**, and the body of the message must contain a line starting with **FREQ** followed by the file or alias to fetch. Other commands, such as **UUENCODE**, **REPLY-TO** and **HELP** are also supported: a full listing of the commands and how they work is in the file *FREQHELP.TXT* included with Internet Rex. This file (*FREQHELP.TXT*) gets sent back to people who send file requests with incorrect commands or who request the help file with the **HELP** command.

Setting *Process email FREQs* to "yes" tells Rex to start looking for these special email file requests. Where it looks for them is determined by the *Matching only through* field. If Rex finds an email file request coming through any of the email addresses you give here, it will process it. If it comes through an address not listed here, it will be treated as a standard piece of email.

Expert mode only If you have the user level set to Expert, you'll have an additional field under *Matching only through: All messages through*. For any email addresses you list here, Rex will treat **all unidentified messages** (that is, all messages which aren't from someone in the nodelist or one of the programs you setup to decode stray messages from) as file request email, even if the message's subject isn't **FREQ**. This can be useful if you want to setup a dedicated file server type email address. Be **very** careful enabling this option: if you set it up for an email address which also receives personal email for you, messages to you (which probably won't have **FREQ** as the subject) will have the file request email help file sent back as a response. In addition, if you've set Rex up to delete **FREQ** messages, the personal email will be deleted before it reaches you, resulting in lost mail.

Getting to know Rex

When Rex does send a file or files back for an email file request, it will use the encoding method you choose in *Default encoding*. Users can change this in their request message using the UUENCODE or MIME commands before requesting their files. If you're sending files through a mailer that splits large messages, you can also specify the number of lines you'd like each message to contain in the *Default chunk size* field. (Setting this to zero will keep Rex from splitting files itself.)

BETWEEN.BAT processing of received mail

Files, paths and programs → *Internet Rex files and paths*

After running Internet Rex to fetch mail, you'll probably want to process the stuff you've downloaded before running Rex to queue and send mail out: that way the files and mail that get generated for other systems get sent out as quickly as possible. Normally, you'd have to do two runs of Rex to work this out: one to fetch the mail and spend time processing it, and another to send it out. Instead, Rex offers the alternative of running a batch file after downloading your mail and before queuing it. That's what the BETWEEN.BAT options in Rex's files and paths menu is about.

If you set *Run BETWEEN.BAT/.CMD* to "yes", and you run Rex so that it's fetching and queuing mail in one session, Rex will check the directory it was installed to for the existence of a BETWEEN.BAT batch file. If it you receive mail during the fetch run and the batch file is there, Rex will run the batch file before continuing to process your outbound mail. This lets you put any commands for mail tossing, processing TIC files or interBBS door game packets into the batch file, process the inbound mail and toss it out again without having to run Rex twice.

Enabling BETWEEN.BAT running will also bring up another option, *Always run BETWEEN.BAT*. Setting this to "yes" will ask Rex to run the BETWEEN.BAT batch file every time it's run, regardless of what command line arguments you gave Rex or whether or not you received any mail. This can be useful if there is a task you have to run before Rex sends files.

OS/2 The OS/2 version of Internet Rex runs the batch file "BETWEEN.CMD" instead of BETWEEN.BAT.

Linux The Linux version of Internet Rex runs the batch file "between" instead of BETWEEN.BAT. Be sure that the permissions on between are such that Rex can actually execute the file when the time comes and remember that Linux is a case sensitive operating system: "BETWEEN" will not get run.

Automatic file bundling and extraction

Files, paths and programs → *Compression programs / Decompression programs*
Node manager → *(node)* → *File bundling*

Many people pay for their internet connections by the minute or by the byte transferred. As a result, squeezing every last byte out of what's transferred becomes important. To

Getting to know Rex

accommodate these sorts of connections, and just to provide an easier “one file sends it all” type of transfer, Rex supports automatic file bundling for files it transfers. Using this, you can have Rex compress all outgoing mail for a given node into one file; at the other end, Rex can automatically extract files from inbound archives making compressed mail transfers nearly seamless. To set all this up, Rex does need to know a little bit about the archivers on your system.

Setting up the compression and decompression programs

Configuring the compression and decompression programs Rex will have to call to create and extract the archives is done in the *Files, paths and programs* menu using the *Compression programs* and *Decompression programs* options. If you want to use a program to bundle outbound files or extract inbound ones, you’ll have to configure it in one of these menus first.

After selecting either choice, you’ll be presented with a list of the operating systems Rex supports and you can choose which operating system you’ll be configuring programs for. You should configure programs in each of the operating systems you’ll be using Rex with: the DOS version will look in the DOS program setup for compression and decompression programs to run, the OS/2 version in the OS/2 program setup, and so on. If Rex doesn’t find a program configured for the OS it’s running under, it will skip bundling or extraction of files.

The *Compression programs* menu will present you with a list of archivers that might be configured on your system and the settings that should be used with Rex. You can add or remove programs from this list as you like.

When adding a new program to the list of archivers, you’ll have to specify a *Tag*, a *Command line* and a *List character* for the program. The *Tag* is a three-letter identifier for the program: when you’re presented with a list of archive formats to create, Rex will list them by their tags. Usually the extension the program uses for its archives is a good choice: ZIP for PKZip, ARJ for ARJ, UC2 for UltraCompressor II, etc.. The *Command line* is combined with the *List character* to create the invocation of the program. When creating bundles, Rex puts a list of the files it wants bundled into another file, and then calls the program with command line options to read the list of files and create the bundle. Rex actually calls the program with the command line

command line **archive name** *list character* **file list**

For PKZip, for instance, this might come out as

```
pkzip.exe -ex -a Archive.ZIP @files.lst
```

where *archive.zip* is the name of the bundle Rex wanted to create and *files.lst* is the file containing the list of files to put in the bundle. So when you add in a new program, you should make the *Command line* such that it will read in a list of files to put in the archive from another file, and the *List character* such that the program will know what the list file is.

The *Decompression programs* menu is a little different. Rather than allowing you to give just any dearchiver, Rex presents you with a list of the archive types it is aware of and lets you specify the program and command line options to give to extract files from that type of archive.

Down the side of the menu is the list of archive types Rex is aware of, along with a special type called “Unknown”. The “Unknown” type will be used when Rex can’t identify

Getting to know Rex

the archive used but knows it has to extract the file. This should be a universal dearchiver type program if you have one. (If you don't, that's fine too.)

The *Command line* you give will be combined with the *Type* and the archive's filename to create an extraction command. The *Type* field tells Rex how to get files extracted into a particular directory: "cd <path>" tells Rex to first change to the destination path, then extract the archive; the remaining choices ("<path> *.*", "*.* <path>", "*.* #<path>", "*.* -d<path>") are just added to the end of the command line. "Default" tells Rex to use the default extraction type for that program. For instance, if the archive to be decompressed were `archive.zip`, the directory to extract to was `d:\inbound` (your inbound directory, perhaps), the *Command line* was `pkunzip.exe -o` and the *Type* was `*.* path`, the command that would get executed would be:

```
pkunzip.exe -o archive.zip *.* d:\inbound
```

Tune the *Command line* and *Type* parameters to match the programs you're using.

Using file bundling

Once the programs for compressing and decompressing files have been setup, you can start adding in file bundling options for nodes. The file bundling information is controlled from the *Node manager* → (*node*) → *File bundling* menu.

If a node is sending you its mail in a file bundle, you can have Rex automatically extract that bundle when it arrives by setting the *Auto-extract incoming* field to "yes". Be careful to do this **only** when the remote is sending you mail in file bundles. Otherwise, Rex will start to extract any inbound file it receives from that node, which could include arcmail bundles, files sent out through file echoes and so on, leaving your inbound directory in a rather messy state.

To have Rex bundle mail for a node into a file bundle, set *Bundle outgoing mail* to "yes". Two new fields will appear: *Archive name* and *Archiver to use*. The *Archive name* is just the name of the archive you want this node's mail bundled into. This should be just a regular filename, with or without an extension. If you don't give an extension, Rex will add one automatically based on the tag of the archiver you choose. For instance, you could set *Archive name* to "bundle". If you used the Zip archiver, Rex would create a file called "bundle.zip" to send this node's mail in.

A problem comes up here though: if you're always sending the same file to the other person, they might start to overwrite each other when the remote is downloading them. To deal with this, you can put #s into the bundle name, and Rex will replace these with a different number each time it's run. For instance, if you set the *Archive name* to "mail####", the first packet sent would be "mail0000.zip", the next would be "mail0001.zip", and the next "mail0002.zip", and so on.

If you decide to use the auto-numbering feature, Rex can let you do a little more adjustment of the file bundles. The *Max size* field will appear: this lets you control the maximum size of the file bundles Rex creates. Leaving this at zero will have Rex create one bundle for each run. If you put it at 500K, Rex would pull 500K worth of files out of the mail queued for the current node and run the compression program. Then it would pull another 500K out and run the program again, creating a different bundle each time, making none of them more than 500K in size.

Getting to know Rex

Finally, you can control the *Archiver to use* for each operating system you run Rex under. You'll be presented with a list of the archivers you have defined for each operating system: choose one, and that will be the archive type Rex bundles files into for this node under that operating system. You don't have to choose the same archiver for each OS, but it is recommended.

Detection of duplicate mail

Connection defaults

Despite our best intentions, setups don't always go as planned. Sometimes, duplicate mail, either arcmail bundles or interBBS game packets or just about anything, can be sent out. Over a standard mailer link, sending the same file twice will result in the second copy being refused. Over Rex, because things are spaced out over the internet, duplicate files don't always arrive in the same session. To fix this, Rex offers duplicate file protection over a period of time: if the same file arrives twice in this time, Rex will delete the second copy.

Two duplicate detection fields are available: *Dupe protection days* and *Dupe protection on all links*. *Dupe protection days* simply tells Rex how long to keep a record of the files that have been received. Rex will check incoming files against this record when looking for duplicates. *Dupe protection on all links* tells Rex what connections to apply duplicate protection to. With this field set to "no", Rex will only do duplicate checking on reliable connections (see the section on [Reliable email connections](#) for more details), because these connections are the only ones which are guaranteed to provide a unique identifier for each file that gets sent. Setting this field to "yes" means Rex will apply duplicate protection to unreliable links as well. It does this by keeping track of the "signature" of each incoming file: its name, and 20 bytes or so of information about the file's contents which will match only one file out of every million billion billion files. (Not perfect, but pretty close.) If a new inbound file has the same name and the same signature, Rex considers it a duplicate, logs it in the log file and deletes it.

Rex's log

Logging options

By default, when you start running Rex, all the logging options will be turned on. That means Rex will log just about anything and everything it does to its log file, in case you need it to figure out problems with the connections, and just to let you know what it's doing. As things go on though, you're not likely to need all the information Rex logs: you can choose which portions of the log file you want Rex to stop logging here. You can also select how Rex will manage its log file: multi-megabyte log files aren't everyone's favourite thing, and with log auto-trimming Rex can ensure you don't have to worry about that.

Each of the lines Rex adds to the log file starts with a symbol to tell you what sort of log entry it is. To enable or disable logging of a particular type of message, just set the field for the log entry of your choice to "yes" or "no". Most people won't need *Debug logging* on all the time: the remaining choices are usually good to keep, but optional nevertheless.

Getting to know Rex

Controlling the size of Rex's log file is just a matter of setting a maximum size in the *Logfile maximum size* field and running Rex with the `-trim` command line argument. You can also have Rex automatically trim the log file after every run by setting *Autotrim logfile* to "yes". If you have a big maximum logfile size or a slow hard drive, this will slow things down a little at the end of Rex's run.

Not for beginners If you have the user level set to something other than "Beginner", you can also decide how Rex's log file should indicate an end-of-line. DOS, in its infinite wisdom, provides two ways of ending a line in a file: the "text mode" carriage return/line feed pair, or the "binary mode" carriage return. You can choose which Rex will use in the *Logging mode* field. If you find the program you're using to read Rex's log runs all the lines together into one big line, you've probably got this in binary mode when it should be in text mode. On the other hand if your program inserts an extra blank line between every line in the log file, you've probably got this in text mode when it should be in binary mode. If you don't have any problems reading the log file, then there's no reason to change this setting.

The queue editor

Queue editor

The queue editor function of the configuration programs lets you see what files have been queued for sending to the various nodes in the node manager. From here, you can snoop around in Rex's queue, change details in the packets queued, pull files out of the queue back to disk or remove mail before sending it in case there was a problem.

After entering the queue editor, you'll be presented with a list of the queue files, each with the name of the person who will be receiving the mail, their netmail address, and some quick details on the queue file itself: when it's due to be resent (if applicable), whether it's been resent and so on. A sample view of the queue editor might look something like this:

The person files are queued for is on the left, followed by the time until it's next resent (or "not sent" if it hasn't been sent yet), and after that a series of letters indicating its queue status:

I	Inbound file (multipart message)
O	Outbound file
S	The file has been sent at least once
R	The file has been queued for resending
Q	The file is still in the queue
F	The file has been expired forcibly

You can select any one of the queue files listed to get more details about it. Doing so will bring you up a menu which looks like this:

Getting to know Rex

```
Internet Rex version 2.00 (Win95/WinNT 32-bit)
(C) Copyright 1997-2000 Khan Software                               Press F1 for help
Queue editor
Joe Sysop (123:123/123.0)                                           4 days 0S---
Jane Sysop (234:234/234.12)                                         3 days 0S---
Jane Sysop (234:234/234.12)                                         4 days 0S---
Roger Sysop (345:345/345.0)                                         4 days 0S---
Roger Sysop (345:345/345.0)                                         4 days 0S---
Roger Sysop (345:345/345.0)                                         4 days 0S---
Tim Sysop (999:999/999.0)                                           4 days 0S---
Tim Sysop (999:999/999.0)                                           4 days 0S---
Tim Sysop (999:999/999.0)                                           4 days 0S---

Sent      Packet has not yet been sent                               Queue number: 135767
Expires   November 15 at 02:16AM
To        Joe Sysop (123:123/123.0)
Contains  4 files (0 acknowledged)
```

Inbound files will be multipart messages that Rex is waiting to complete. These will have *Resend* and *From* fields. Outbound files are files which Rex is either waiting to send for the first time, or which Rex is holding until the remote node acknowledges receipt of the file. These will have *Sent* and *To* fields. On the right side of the screen, you'll see this queue file's *Queue number*: this lets you identify which of the files on disk Rex is looking in for the information you're looking at.

The most common type of queue file you'll see in Rex's queue by far will be files that are waiting for acknowledgement from other nodes. If you choose one of these files, you can change the queue file so that Rex thinks it has been acknowledged (by choosing the *Expires* field and answering "yes" to the request to expire it now), or you can modify the contents of the queue file: delete files from it, move them back to the hard drive, or change their filename. Do this by selecting the *Contains* field.

The contents of the queue file are displayed as a list of files: filename first, then the size of the file's entry in the queue file (usually the same as the file's size, unless it's been split into multiple parts), then the file's 32 bit CRC, it's confirmation number, and finally a few letters to tell you whether the file has been sent or not (S), acknowledged (A), resent at least once (R), is a receipt file (T) or is one part of a multipart file to be sent (P). The details are just there for a quick reference: anything you want to change has to be changed by selecting a file entry to edit with the **enter** key. This will bring up a new menu:

Getting to know Rex

```
Internet Rex version 2.00 (Win95/WinNT 32-bit)
(C) Copyright 1997-2000 Khan Software                               Press F1 for help

070B0112.896                1535 295B726C 3D704EC1 S----
070F0107.800                412  87808D2B 5B815D64 S----
070B0112.898                822  6B1EDCEB DAC863C6 S----

Filename :070B0112.896                Sent Yes
Location :Stored in queue              Acked No
Special  :                             Parts 1
Size     :1535 bytes                   CRC-32: 295B726C Confirmation number: 3D704EC1

Sent      Packet has not yet been sent      Queue number: 135767
Expires   November 15 at 02:16AM
To        Joe Sysop (123:123/123.0)
Contains  4 files (0 acknowledged)
```

Most of the information presented here you don't have to worry about. (For more technical definitions of all of the fields, see the technical manual.) Some of the information you can change though. If you'd like to assign a new name to the file being sent, you can change the *Filename* by selecting the field and typing in a new name. If you'd like to have just this file resent, you can change the status of the *Sent* field for this file. Similarly, you can mark or unmark a file as being acknowledged by the remote by toggling the value of the *Acked* field. Finally, if you'd like to remove the file from the queue and put it back on your hard drive for modification, you can do so by selecting the *Location* field. Returning the file to the disk will take the file out of the queue and put it in the inbound files directory of your frontend mailer.

There are more things you can do with the queue editor, more than can be covered here without going into some gory details about Rex's queue structure. If you'd like more information about the queue editor, look in Rex's technical manual, or refer to the online help.

Reliable email connections

Connection defaults

Node manager → (*node*) → *Connection information*

Despite the best-laid plans of the internet administrators, the internet itself is not a reliable connection. Mail servers going up and down, changing mail along the way, FTP servers with different protocols or slightly disconnected adapters can all result in the data you send not being what the other end receives, if it receives it at all. Losing mail in a Fido network can result in more than a few people being a little annoyed, especially since modem connections, while not particularly fast or cheap, are at least fairly reliable. Internet Rex offers a way around this: a method of ensuring that the mail you send is what the other end receives. This feature is available when talking to a number of different programs, not only Internet Rex, but also Allfix and TransX. It works by including a manifest in each email or FTP connection it sends, a listing of all the files that are supposed to have been transferred and a signature for each of them to make sure that the file received is the same as the file sent.

Getting to know Rex

Nodes which receive this manifest check it against the files they've received, and if everything matches, an OK message is sent back; if there's a problem, a request for a resend is sent back instead.

Reliable email connections are configured in two places in Rex. The first is at the node level: for each node in the node manager you can define whether you want the connection to be reliable or not, and how you'd like the connection to work. This is done in the *Connection information* menu of the node manager.

Node level options

Users with a user level of Intermediate or Beginner will have just one field to set to establish a reliable connection: *Reliable connection*. Set it to "yes" to have Rex set the connection details up for you, "no" if you don't want the connection to be set up as reliable. Note that if you set this to "yes", the node on the other end **must** be using software that understands reliable connections for the setup to work. At the moment, this includes only Internet Rex, Allfix, TransX and any SEAT compatible mailers. If you try to setup a reliable connection to a node that doesn't support them, you will end up sending and resending files to them because they never send back receipts acknowledging they received the files correctly.

Advanced or Expert users will have a little more liberty in setting up reliable connections. There are five fields to do this with: *Send acknowledgements*, *Resend request delay*, *Accept resend requests*, *Purge delay* and *Auto-resend unacknowledged mail*. The first two and last three are linked.

Send acknowledgements tells Rex to send back receipts OK'ing or requesting resends of mail that has been received. You can set this to "yes", "no" or "batch". "Batch" is equivalent to "yes", except that instead of sending back one message for each acknowledged file, it sends all the acknowledgements for the files it receives in one session in a single file. This will cut down on the number of messages Rex has to send. This setting should match what the remote has setup for their equivalent of *Accept resend requests*. The *Resend request delay* tells Rex how long to wait before requesting a resend of incomplete multipart mail. (Incorrectly received mail will have a resend request sent immediately.)

Accept resend requests tells Rex to keep a copy of mail sent to this node in its queue directory until it's received an OK from the other node. How long that mail is kept before Rex gives up on trying to send it is determined by the *Purge delay* setting: after this time Rex will either delete the mail, or if you turn on *Auto-resend unacknowledged mail*, will resend it. The number of times it resends it is set in the *Connection defaults* menu.

A good set of values for the five reliable connection fields is:

<i>Send acknowledgements</i>	Batch
<i>Resend request delay</i>	32 hours
<i>Accept resend requests</i>	Yes
<i>Purge delay</i>	96 hours
<i>Auto-resend unacknowledged mail</i>	Yes

Getting to know Rex

Global connection options

Because of the difference between connections, the above settings are done on a node by node basis. (You would get a receipt back much faster from someone who checks their mail every few hours than from someone who checks it once a week.) There are some values for reliable connections that apply the same to every connection, and these are setup in the *Connection defaults* menu.

Although reliable connections generally ensure that mail will arrive, eventually, there are some cases where Rex just has to give up and stop try to resend the files. (Someone changing email addresses without notifying you for instance.) For this reason, there is a *Maximum resends* field: after this many resends, Rex gives up trying to send the file and just deletes it from its queue. You can set this to anywhere from 1 to 8. Similarly, there are some cases where Rex just won't be able to receive a multipart file: if the remote has deleted the file because of maximum resends, for instance. In this case, you have to give Rex a maximum amount of time to keep a partially completed multipart file around before giving up: this is done in *Partial purge delay*. How this setting is applied is set by the *Purge known partials* field. If the partially completed file is from a known node (one in the node manager), you can have Rex keep the file indefinitely by setting *Purge known partials* to "no". In that case, only partially received files from anonymous nodes will be deleted after time. Otherwise, all partial files fall under the chopping block eventually.

Rex being the bright puppy that he is can also acknowledge files from nodes not in the node manager, by sending receipts back to the person listed in the From: address of the email message sent. If you'd like to enable this feature (it's useful while testing new links with nodes using a reliable connection), set *Default send acknowledgements* to "yes". Enabling this also requires you to give a delay for requesting resend of partially received multipart files from anonymous nodes: that can be entered in the *Default resend delay* field.

Secure and encrypted mail

Node manager → (node) → *Connection information*

For many Fido networks, the question of security is an important one. The last thing people want is someone spamming your network with loads of fake mail, or reading someone else's netmail. The open nature of the internet makes this even more of a concern, as faking email is not a hard thing to do. Rex has a number of solutions to these problems, some common to standard Fido file transfers, some a little more advanced.

Many people set up session level passwords in their frontend mailers to authenticate the person on the other end: this way someone can't just connect with a mailer and arbitrarily transfer mail, they have to have a password in place. Internet Rex allows you to do something similar with mailers that support this feature. The *Session password* field can be used to setup a password to authenticate mail transfer between two SEAT level 3 mailers (at the moment, this only includes other copies of Internet Rex). Rex also uses this field as the session password sent during BinkP connections. The password can be up to 30 characters long and contain any symbol you like. It's compared against the password the remote uses case sensitively: "PASSWORD" would not be the same as "password".

Getting to know Rex

The *Packet password* in Rex is similar in function to the *Session password*. The .PKT files that Rex creates when exporting netmail allow you to insert a password up to 8 characters long into them for authentication. The remote's mail tosser would check this and passwords that didn't match would result in the .PKT being flagged as insecure or corrupt. The password here is compared without regard to case: "PASSWORD" and "password" would be considered the same.

If you're looking for real security in transferring mail, the above two methods provide some assurance that the mail you receive really is from the person that claims to have sent it, but it provides no protection against other people examining the data. If you'd like to be sure that no one will read the messages you send through Rex except you and the person who is supposed to receive them, you can add a further level of security to your mail by encrypting the files. Encryption garbles files as they're sent so that only someone who knows how they were encrypted and the password they were encrypted with can decode them and read them. If you'd like to setup an encrypted link, the person at the other end must be using a SEAT level 4 mailer that supports encryption. (Currently, only Internet Rex falls into that category.) If both people are using the right software, you enter the password you'd like to use into Rex's *Encryption password* field and choose an *Encryption method*, either "Blowfish" or "S-Coder". Blowfish is a very secure algorithm for encrypting files which should ensure that whoever wants to try to read your mail will need a supercomputer and a few million years to figure out what it says. It also takes a fair bit of processor time to do its encryption, at least in comparison with the other encryption method offered. S-Coder is a less secure algorithm than Blowfish: someone who knows what they're doing could probably crack the password on your mail given a fast Pentium computer and a few days to process it, but it doesn't take as much time to encrypt the mail. In order to ensure that someone using brute force attacks doesn't guess the password for your mail immediately, Rex requires that encryption passwords be at least 8 characters long.

When decrypting encrypted mail, both nodes must have **exactly** the same password and encryption method in place, otherwise the mail will end up looking like a garbage file. Exactly in this case means that every character must be the same: "PASSWORD" on one end and "password" at the other will not work.

With all these features available, there can be no question that a secure connection can be setup between two nodes. Even so, Rex allows you to define your own criteria for a secure connection or not with the *Secure connection* field. Setting this to "yes" tells Rex that this connection is secure, regardless of what passwords you've put in place. A securely connected node will have its files saved in the secure inbound directory of your mailer instead of the regular inbound, and will have access to files described in the secure alias and directory lists of your file request setup. Setting *Secure connection* to "no" tells Rex that the node should be treated like any other.

Transfer statistics

Node manager → (node) → *Statistics*

So you've been running Rex for a while, and you're curious as to how much mail you've sent to someone over the last little while.... Or you run an interBBS league through

Getting to know Rex

Rex and are wondering how long it's been since someone sent you a packet.... Or you just plain want to see lots of numbers and dates and stuff.... To help you in your quest, Rex keeps statistics on all the nodes in the node manager.

Rex's statistics include the last time Rex received, queued or sent mail for a node, how many files were sent or received and how many kilobytes were sent or received. These are kept track of along with the date you last reset all these values. To see a node's statistics, just select the *Statistics* option from the node editor. If you want to reset these to zero, hit **return** in the statistics screen. To close the window, hit **escape**.

Rex can also generate files that display all the statistics for all the nodes in a compact form. You can access these functions by running the configuration program with the `-stats` or `-dates` command line switches. See [Command line options for the configuration program](#) for more details.

Advanced node setup options

Node manager → (*node*)

One option for node setup that wasn't discussed before was the use of the *Node active* field. You can set this to "yes", "no", "receive files only" or "send files only". When in "receive files only" mode, Rex will check to see if the node has sent any files, but won't send or queue any mail for this node: "send files only" mode is the opposite of this, where no mail is downloaded, but mail present is queued up and sent. Setting *Node active* to "no" stops Rex from processing anything for this node, either inbound or outbound. Setting it to "yes" leaves the node running normally.

If you have your user level set to Advanced or Expert, there are a couple of extra options available for configuration when you go to manage a node.

Crash all mail is useful for when you're running Rex in daemon mode. If you have this set to "yes", Rex will treat any mail for this node as crash mail. If you happen to have the *Send IMM/Crash mail* field (in *Rex's behaviour*) set to "Send immediately", this means Rex will send any mail for those nodes as soon as it sees it: it won't wait for an event to send it. This is great for moving mail through your system quickly.

The *Mailbox directory* can be used to have Rex send all files from a particular directory to this node. If you give a directory here, any file Rex finds in that directory will be sent to the node you're setting up. Note that Rex does nothing except send the files from the directory and delete them when it's done. Many have tried to use this to send files from Binkley outbound directories for other domains: this **will not work**. If you have more than one domain setup in Binkley, you can have Rex process the mail for that domain by adding the domain's outbound and the zone it's linked to to Rex's domain setup in *Address manager* → *Domains*. (See [Domains](#) for more details.) Many have also set this to their netmail directory incorrectly. Again, **don't do this**. A mailbox directory and a netmail directory aren't the same thing. Configure your netmail directory in *Files, paths and programs* → *Frontend mailers* → *Netmail directory*.

The *Netmail bundled to* option lets you specify what sort of .PKT file Rex will put netmail for this node into. "Type 2+" should work nicely for most connections. Note that as

Getting to know Rex

of yet, there are very few mail tossers that support the “Packet 2000” packet type: only select this if you’re sure it will work with your remote’s mail tosser.

RexFix remote management

Node manager → (*node*) → *Connection information*
Rex’s behaviour

In the *Rex’s behaviour* menu, there are two options for controlling a feature of Rex called RexFix processing. RexFix processing allows Rex to process requests from nodes in the node manager to change their setup options. This lets nodes pause mail sending for a while (when they go on vacation for instance), change email addresses, transport methods, site addresses and so on, all without bothering you with the setup details. They do this by writing messages to RexFix in a particular format. (See the file REXFIX.TXT included with the standard distribution for more information on the message format and commands available.)

Whether or not Rex processes these requests is determined by the value of the *RexFix processing* field: “enabled” means any node can send RexFix messages and have Rex process them. “Disabled” turns off RexFix processing: Rex will ignore any RexFix messages and simply post them to netmail like any other piece of mail. If you decide to enable RexFix processing, the *Delete RexFix messages* will determine whether Rex deletes the inbound messages once it has processed them, or whether it will post them to netmail.

Each node has a *RexFix password* field available in the *Connection information* menu of the node manager. If there is no password given here, Rex will process any RexFix request that appears to be from that node. However, if a password is given here, Rex will only process requests that have a matching password in the subject line of the message. This lets you make sure other people aren’t changing your nodes’ setups: remember that email is easily faked. Each node also has flags which let you decide whether the node will receive RexFix help or notify messages when you create these from the command line. Setting *Send help messages* or *Send notify messages* to “no” will prevent Rex from creating messages for these nodes from the command line unless you specifically tell Rex to. Use this for masking out uplinks when sending out notify messages. For example, if 1:342/0 was your uplink, and you had a number of nodes in net 342, you could send notify messages to only them by setting the uplink node’s *Send notify messages* to “no”, and specifying `rex -notify 1:342/*` on the command line.

Command line options for the configuration program

The configuration program has a few command line options which you may find useful when generating statistics, nodelists or playing with the colour configuration.

```
-stats [reset]
```

This option will generate a statistics file in STATS.TXT. It basically dumps all the statistics Rex has gathered on all the connections in the node manager to that one file, listing the

Getting to know Rex

number of files sent and received, the number of kilobytes sent and received, and the time the statistics for each node were last reset. You can optionally tell Rex to reset the statistics for all the nodes in the node manager by adding the `reset` switch after the `-stats` switch.

`-dates`

The `-dates` switch is another statistics option for Rex. Instead of listing the number of files and kilobytes transferred, this option lists the dates that Rex last received, queued and sent mail for each node in the nodelist. The statistics are dumped to the files `DATES.TXT`.

`-nodelist`

This can be used to dump the contents of the node manager to a text file. Every node listed in the node manager will be written out to the file `NODELIST.TXT`, including their addresses, email, FTP or BinkP setup, connection passwords and so on. Useful for creating a text backup of your `*.REX` files in case they're lost or damaged.

`-queue`

This lets you skip the start-up menu of the configuration program and takes you straight to the queue editor.

`-original`

If you were playing with the colour configuration portion of Rex and accidentally setup a colour scheme that hides some important options, or you just want to reset back to the original scheme for a while, giving the `-original` switch will tell Rex to use the original colour scheme this one time it's run.

`-monochrome`

Like `-original`, `-monochrome` tells Rex to use the monochrome colour scheme for this run of the configuration program.

`-docs [lines per page [left indent [page width]]]`

Rex includes both this user manual and a technical manual that details every configuration option in Rex and what it does. The technical manual isn't included in the standard distribution, but it can be created by running the config program with the `-docs` switch. You can also configure how the manual will be printed by providing the number of lines you want on each page, the number of spaces you want on the left side of each page and the width of the page in characters. For example, `-docs 67 6 68` creates the technical manual with 67 lines per page, 6 spaces of indent on the left of the page and 68 characters worth of text. (This makes the total line width 74 characters.)

Internet Rex's servers

Starting with version 2.00 of Internet Rex, running Rex in daemon mode allows you the option of running built-in servers, also known as daemons. You can use these to provide services to people using your machine. The FTP server can be used for sharing files on the internet or accepting FTP uploads from downlinks; the POP3, SMTP and finger servers let you host your own internet email addresses; the BinkP server lets you receive inbound BinkP connections; and the ident server lets you provide verification for other internet programs (IRC clients, for example).

Requirements

In order to run the various servers Rex includes, your setup will require:

- A full time internet connection.
- Rex must be running in daemon mode.

Rex must be run in daemon mode to ensure that it stays around long enough for the servers to be accessible. The full time internet connection isn't necessarily required, but without it, your servers will be inaccessible when the connection is down. A lack of a full time connection can also cause problems for some of the servers: for example, if the connection is down while the SMTP server is trying to deliver mail, that mail may end up being sent back to the sender instead of being properly delivered.

If your copy of Rex isn't a registered server version, you'll find that although the servers still work, they are somewhat limited: you can only configure two FTP users and two mail users in unregistered versions. The BinkP and ident servers work the same in both registered and unregistered versions.

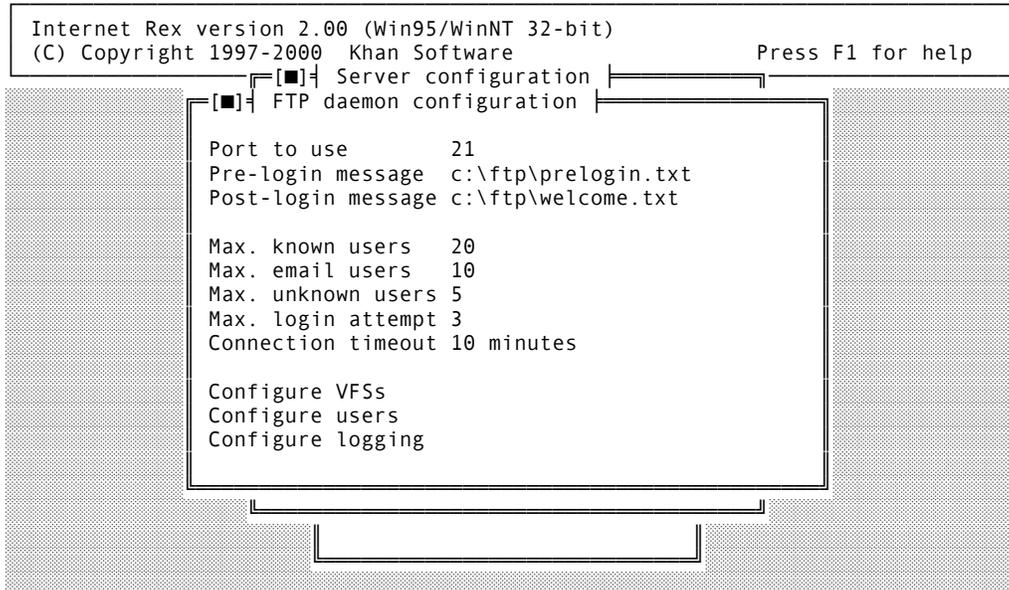
The FTP daemon

Server config → *Configure (FTP daemon)*

Internet Rex's FTP daemon allows users on the internet access to the files on your hard drive in a controlled manner. Anonymous users can be setup to allow anyone to download particular files, or you can setup passworded users with greater access. For each user, you can setup upload/download ratios, time online limits, permissions for the directories and files on the server and more.

Heading to the configuration menu, you'll be presented with this screen:

Internet Rex's servers



This lets you configure the basics of how the FTP daemon will work. The daemon must first listen on a TCP/IP port on your computer: the usual *Port to use* is 21. If you choose a different value for this field, you'll have to inform people connecting to your site to use that new port, otherwise their FTP programs won't be able to reach the server.

Once someone has connected to your site, they'll be presented with the *Pre-login message* (a text file of your choice, or blank if you don't want a pre-login message) and then the login prompt. After they login, they get the *Post-login message*.

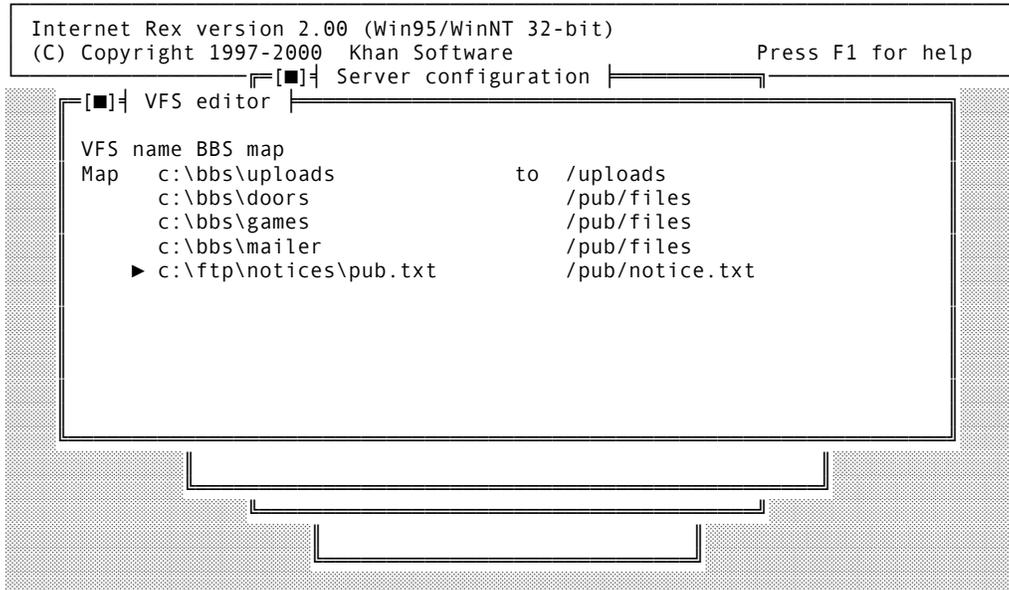
Whether or not the login succeeds is determined not only by the user's password but also by the limits you put on simultaneous FTP users. You can set a maximum for *Known users*, *Email users* and *Unknown users*. Known users are users with a password; email users have to enter their email address as a password; unknown users don't have to enter a password; you can configure the user type for users in the user setup menu.

If the user has a password, they'll have a limited number of chances to get their password right before Rex drops the connection: the maximum number of tries is set in the *Maximum login attempts* field. Once they're logged in, a timer is setup to log them out if they're inactive for a certain number of minutes, the amount set in the *Connection timeout* field.

Setting up virtual file systems (VFSs)

Once a user has logged on, what they'll be able to see on your FTP site is determined by what virtual file system (VFS) you've given them. A VFS is a way of creating a whole different view of your hard drive for your FTP users. With it, you can map directories on your hard drive to virtual directories on your FTP site, merge multiple directories together into one huge directory, link files to other files and so on. To add a new VFS, select *Configure VFSs* from the main FTP daemon menu and hit the insert key. A sample VFS is presented here.

Internet Rex's servers



The first thing the VFS is given is a *VFS name*. The name identifies the VFS when you go to add in users later on. The rest of the VFS consists of maps: links between files or directories on the hard drive and files or directories on the FTP site.

The first map links the directory `c:\bbs\uploads` to `/uploads` on the FTP site. Looking at the VFS from the side of an FTP user then, if they changed to the `/uploads` directory they might see the contents of `c:\bbs\uploads`.

☞ Note that the user only might see the contents of `c:\bbs\uploads`. VFSs only define what the underlying setup of the FTP site is. They don't define whether a user has permission to read / write / list / delete etc. files or directories on the FTP site. Permissions are set on a per user basis in the user setup menu.

The next map takes `c:\bbs\doors` and maps it to `/pub/files`. Note that it wasn't necessary to define `/pub` as mapping to something: the FTP daemon will create a virtual, empty directory called `/pub` with the subdirectory "files" in it. The two maps afterwards also map to `/pub/files`. This means that when a user lists the contents of `/pub/files`, they'll see the contents of the directories `c:\bbs\doors`, `c:\bbs\games` and `c:\bbs\mailer` in one big list.

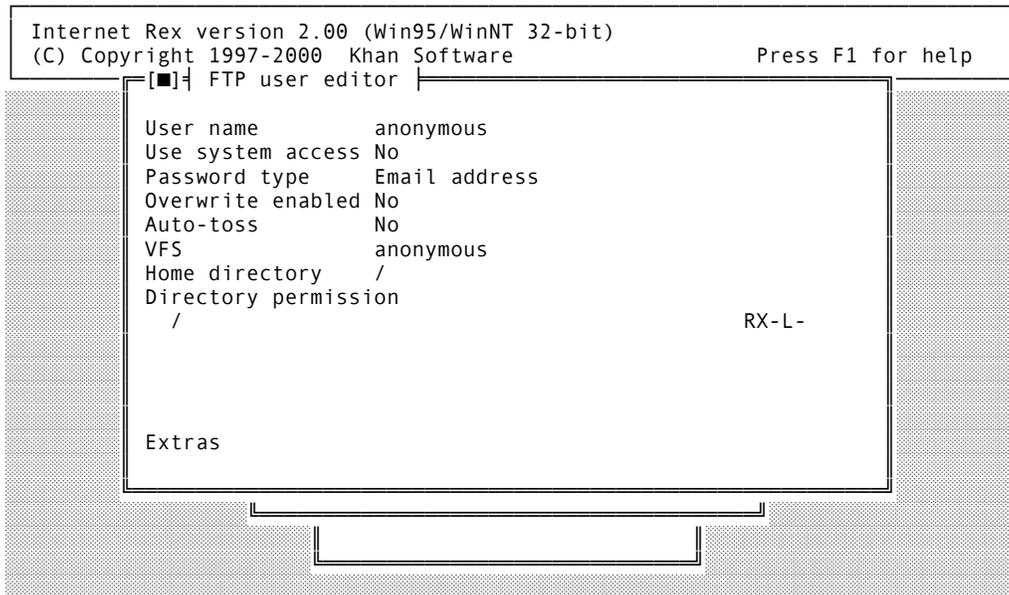
Maps don't have to be limited to directories. The fifth map, for example, maps the file `c:\ftp\notices\pub.txt` to `/pub/notice.txt`. When a listing of `/pub` is made, there will be a file called `notice.txt` there with the same attributes as `c:\ftp\notices\pub.txt`: if that file is downloaded, the contents of `c:\ftp\notices\pub.txt` will be sent.

Other tricks of VFS setup can be found in the online help.

Internet Rex's servers

Adding a user

With a VFS setup, you can now add a user to the FTP daemon. As before, choose *Configure users* from the FTP daemon main menu, and hit insert. You'll be presented with this screen:



Every user must have a *User name* to login with. Whether they need a password or not is determined by the *Password type* field. If set to "Encrypted", you can enter a password in the password field below and that password will be encrypted and stored. The user must then enter the same password to login. If set to "Email address", the user only has to give their email address to login. If set to "None", the user can login without giving a password or email address. Finally, you must choose a *VFS* for the user. This determines what the FTP site will look like for them.

When a user logs in, they will be put in their *Home directory* to start out with. (The home directory must be a valid directory in the user's VFS. Don't specify this as a directory on your hard drive.) What they'll be able to see and do on your FTP site will be determined by the *Permissions* that you give them.

Each permission entry is a directory on the FTP site and a set of actions the user can perform. There are five different permissions:

- Read permission grants the user the permission to download the files in a particular directory.
- Write permission lets the user upload files to that directory.
- Execute permission allows the user to change to that directory.
- List permission means the user can list the contents of a directory.
- Delete permission allows the user to delete files or subdirectories of the directory and to overwrite files in that directory.

Any permissions for a particular directory also apply to every subdirectory of that directory unless you specifically exclude them further down the list of permissions. So if you gave someone read, execute and list permission to /, every directory on the FTP site would be

Internet Rex's servers

available for them to change to, read and download files from. If you then changed permissions to /pub/files to nothing (no permissions), the user could still read, change to and download files from every other directory on the FTP site, but /pub/files would be off-limits to them.

Linux Since Linux has permissions built into the file system and users/passwords built into its logins, the FTP daemon can be configured to use those instead of what you enter here. This in effect changes the FTP daemon to act like a standard Unix FTP daemon. Set the Use system access field to “yes”, and the daemon will take the user name you give and use that user's home directory, password and permission information to determine what they see on the FTP site.

If you plan to have anonymous FTP users under Linux, you must have a user with the login **guest** in your Linux password file. Rex will use that user's user ID, group ID and permissions to determine what anonymous users have access to. If **guest** doesn't exist, anonymous users won't be allowed to log in.

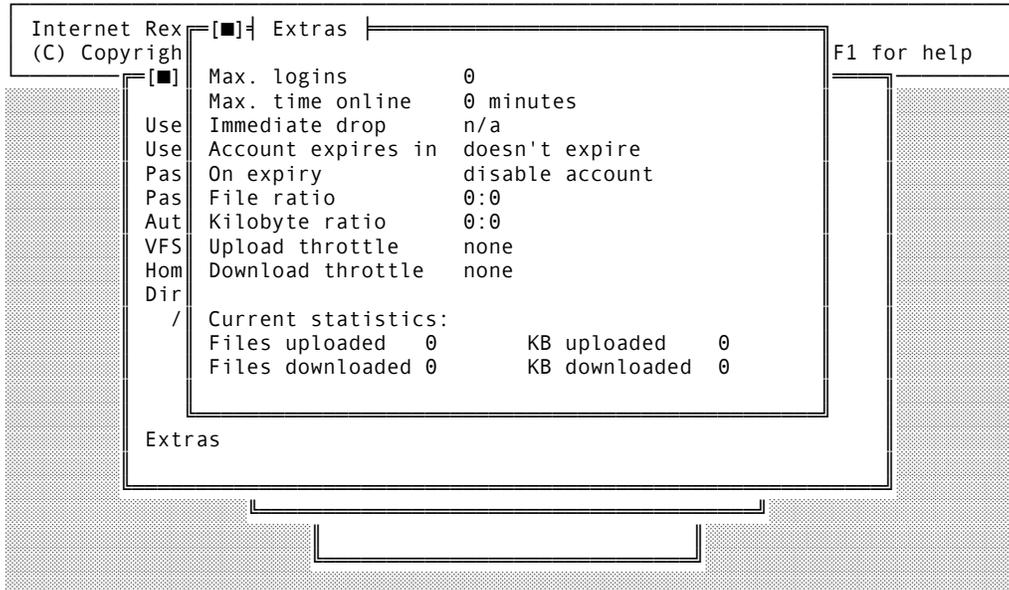
Additional FTP features

Beyond the usual FTP user stuff, you can setup extra information for users to give them file ratios, time limits and so on.

The first of these extras, and the most useful if you're planning on using Rex's FTP daemon to deliver mail to downlinks, is the *Auto-toss* feature. When this field is chosen, you'll be presented with a list of the users in the node manager who are setup to toss mail to or from a directory. When you link an FTP user with a node manager user, Rex will run the toss to/from directory commands for the node manager user when the FTP user logs on or off. For example, suppose FTP user “Downlink” is the account you've given to your downlink 1:342/806.22. You setup 1:342/806.22 in the node manager as a toss to/from directory node and give “Downlink” permission to upload and download to 1:342/806.22's upload and download directories. Then you link 1:342/806.22 to “Downlink” in the auto-toss field and when “Downlink” logs on, mail for 1:342/806.22 will be put in his download directory, and when he logs off, mail from his upload directory will be put in your inbound.

Additional features, like file ratios, are configured by choosing the *Extras* field at the bottom of the user setup menu. You'll get this screen:

Internet Rex's servers



Here you can set the maximum number of times this user can be logged in at once (*Max. logins*), the maximum amount of time they can remain connected (*Max. time online*) or the number of days before their account expires (*Account expires in*). (Setting any of these values to 0 will disable the restriction.) If you've set the user to have a maximum time online, you can determine whether they'll be dropped with extreme prejudice (any uploads or downloads terminated, booted immediately) or allowed to complete their transfers: set *Immediate drop* to "yes" to have them booted right away, "no" to be generous. Similarly, if the account is going to expire, it can either be deleted right away or simply disabled so that the user has anonymous access. (Use the *On expiry* field to do this.)

If you have a limited amount of bandwidth available to you, you can limit the file transfer speed for particular users using the *Upload throttle* and *Download throttle* fields. Entering a value here will restrict that user's uploads or downloads to at most the speed you give, though this is still done on a per connection basis.

Each user can also be given an optional *file* or *kilobyte ratio*. This is specified in terms of uploads to downloads: for example, a 1:8 file ratio would mean the user must upload one file for every 8 downloaded. A summary of the number of files and kilobytes the user has transferred is presented in the *Current statistics* section. These numbers can be changed if you like. For email or unknown users, the statistics only represent what was transferred the last time the user logged on.

FTP daemon logging

The FTP daemon offers two log files for keeping track of the activities of FTP users on your system, both accessible through the *Configure logging* choice off the FTP setup main menu.

Transfer logging, the first option presented, allows you to log what files a user uploads and downloads, and when they logged on. This information can be viewed while Rex is running using the F9 history key at the FTP daemon's screen, or can be imported into most

Internet Rex's servers

spreadsheet programs. Without transfer logging enabled, no transfer history will be available from Rex's FTP window. With transfer logging enabled, you must specify a *Log filename* to keep the details in, the *Number of files to log* and the *Number of lines to show*. The number of files to log tells Rex how many of the files transferred in the session it should remember. If you set it to 10, the last 10 uploads and the last 10 downloads from each session will be logged. The number of lines to show controls how many lines of the FTP daemon history will be displayed when you browse the FTP history in Rex itself.

Standard logging logs the minutiae of the FTP daemon as they occur. If you enable this, you'll have to provide a *Log filename* and enable or disable the various logging options to control what level of detail you want. The options and their data are:

- *Uploads/downloads*: Adds an entry each time an upload or download is attempted. Says what the name of the file is and how large it is.
- *Logins/logouts*: Adds an entry each time someone attempts to login or logout of the FTP server. Keeps track of their IP address, the login name and password used and when the attempt was made.
- *Create/remove directory*: Adds an entry each time a directory is created or removed.
- *Renames*: Adds an entry every time someone renames a file.
- *All commands*: Logs, word for word, the commands sent to the FTP server. Useful for debugging problems users might be having with the server.
- *Mask passwords*: If enabled, this prevents users' passwords from showing up in the log entries. Instead they'll appear 'x'ed out.

The mail daemons

Server config → Configure (POP3 daemon)

Server config → Configure (SMTP daemon)

Server config → Configure (finger daemon)

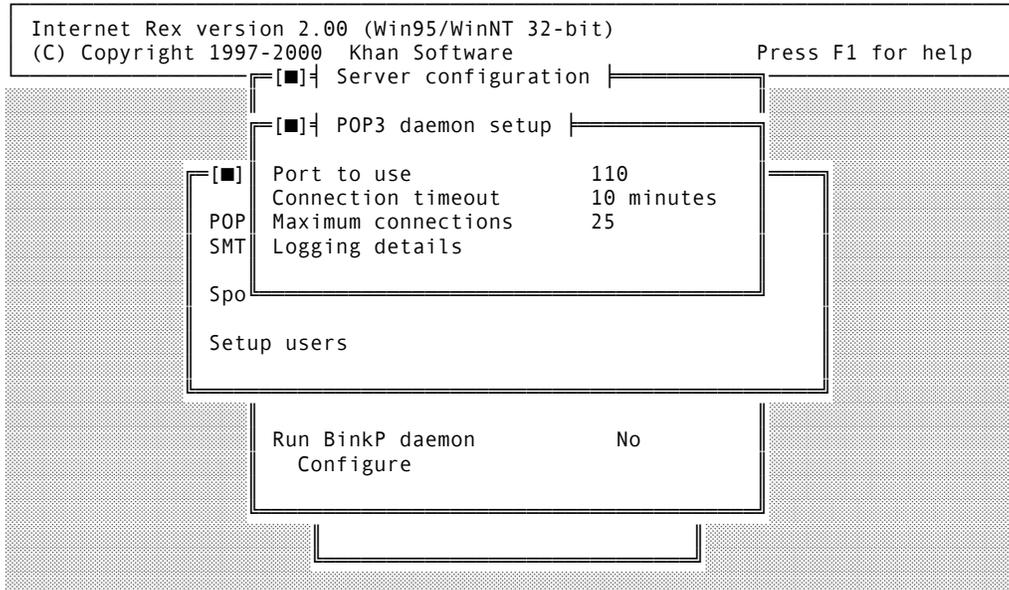
The mail daemons included with Internet Rex allow you to host email on your machine. The SMTP daemon is used for delivering and receiving email, the POP3 daemon is used to provide any users you might setup a way of downloading their mail, and the finger daemon is an easy way for people to find out if they have new mail waiting without logging into the POP3 server. The SMTP daemon in particular allows you to setup mailing lists, mail robots, email aliases and hosted subdomains.

Configuring the various mail daemons is broken into five parts: the POP3 daemon, the SMTP daemon, the finger daemon, the mail spool setup and the actual users themselves.

POP3 daemon setup

The POP3 mailbox server is probably the most common way of allowing users to pick up email. Because it's just there for picking up mail and deleting messages when the user asks it to, there's not much to setup. The POP3 setup screen looks like this:

Internet Rex's servers



Users wanting to pick up mail connect to the server on a TCP/IP port: the standard *Port to use* is 110, but you can specify a different one if you like. Keep in mind that doing so may make things a little tough for your users. The daemon can handle hundreds of users at once, but the more users there are logged on, the more drain there will be on your system, so you may want to limit the *Maximum connections* to something more reasonable. If the maximum is reached and another user tries to pick up mail, they'll simply be told to try again later. To avoid users holding a connection open for no reason, or to deal with users who've lost their connection to the internet suddenly, an inactivity timer is set for each user who logs on. If they don't do anything for a certain number of minutes, the POP3 connection is dropped: the number of minutes to wait is determined by the *Connection timeout*.

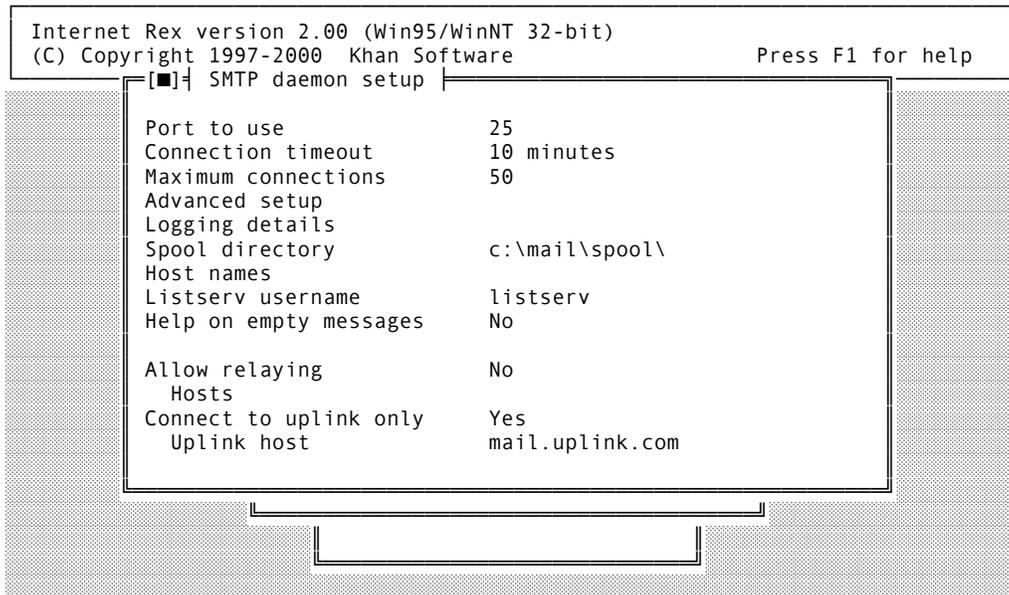
You can set the POP3 daemon to keep a log of its activities if you like. Choose *Logging details* and enter the file you want the log written to in the *Log filename* field.

- *Connection details* gives the IP address and time that a connection is made to the POP3 daemon
- *Login details* gives username and password information or failed login information
- *Messages retrieved* and *Messages deleted* add an entry when a user downloads or deletes a message
- *Errors* adds an entry when an error is encountered while fetching a user's mail or starting up
- *All commands* logs the actual commands the user sends to the POP3 daemon – useful for debugging purposes.
- *Mask passwords* prevents passwords users give when logging in from appearing in the log.

Internet Rex's servers

SMTP daemon setup

The SMTP daemon is the real workhorse behind Internet Rex's mail system. It's what delivers and receives email messages for you, and as such is a little more complicated to setup.



As with the POP3 daemon, the SMTP daemon has a standard TCP/IP *Port to use* (port 25), a *Connection timeout* and a configurable *Maximum connections*. As before, running your SMTP daemon on a port other 25, while possible, will mean that most of the people trying to send mail to your server will be unable to reach it, even if you tell them the port you're running on – as such, it's inadvisable to change it. Also as before, while hundreds of simultaneous SMTP connections can be handled, the more of them there are active, the more strain will be placed on your system, so it's best to keep the number reasonably low. With SMTP connections, it's best to keep the *Connection timeout* fairly high. Email is one of the most heavily used systems on the internet, and this often means that mail servers sending mail to you can be quite slow because of the load they're handling. A longer timeout gives a better chance of receiving mail on the first try if the connecting server is especially busy.

The most important thing about the mail server is probably the *Host names* field. Here you enter the machine names the server owns. For example, if you wanted the server to receive mail for users @somewhere.com, one of the host names you enter should be somewhere.com. The first host name you give will be the primary host name: this will be used to represent your machine when the server connects to other SMTP servers. Note that just entering a host name here doesn't automatically give you control of that domain: your machine must be registered on the internet with that name in order for mail to be delivered to you correctly.

When a message is received from another SMTP server, it's placed in a pair of files in the *Spool directory* until it can be properly delivered. Usually, the message will simply be moved from there to the mailbox of one of your users. If the message turns out not to be for one of your users, it will stay in the spool directory until it can be properly delivered. This can

Internet Rex's servers

take as little as a few seconds or as long as 5 days depending on where the message is going and whether the server that's supposed to receive it is responding or not. Which server the daemon will try to connect to to deliver messages is determined by the *Connect to uplink only* field. Set this to "yes" if you want Rex to use someone else's SMTP daemon to deliver mail – your internet provider's SMTP daemon, for example. You then have to give the machine name to connect to in the *Uplink host* field. If *Connect to uplink only* is set to "no", Rex will try to deliver the message directly to the recipient's SMTP server: this can result in faster mail delivery, but will mean more messages are sent when a message with multiple recipients is delivered. If you want to use this method of delivery, you must provide the machine name or IP address of your network name servers in the *Name servers* field.

You can configure whether you want the SMTP daemon to accept mail from other servers to users who aren't on your system by setting the *Allow relaying* field. Relaying is when one SMTP server delivers another's messages because the first doesn't know how to reach the user, or doesn't want to try. Setting this value to "yes" will allow any SMTP server anywhere to use your server as a relay host. Be careful using that setting: spam email is usually delivered by unsuspecting mail servers who've allowed anyone to use them as a relay. Setting it to "no" will disable relaying of messages: this will ensure you never end up delivering spam, but it also means machines on your local network won't be able to use the SMTP server you've setup. To fix this, there are two in-between settings: "These hosts only" and "All but these hosts". Each lets you give a list of machines (IP addresses, machine names or wildcards): "These hosts only" permits relaying when the connection matches one of the entries in the list; "All but these hosts" permits relaying to any machine except ones in the list.

Listserv username and *Help on empty messages* deal with the SMTP server's processing of listserv requests. Listserv requests are a way of allowing people to join or leave mailing lists you setup by simply sending a message. The *Listserv username* identifies the user listserv requests must be sent to to be processed. (It shouldn't be entered in your list of users later on.) *Help on empty messages* specifies whether a help message will be sent back if the listserv can't find any valid commands in a message it's told to process.

As with the POP3 daemon, the activities of the SMTP daemon can be logged to a file. Choose *Logging details* and enter the file you want logs written to in the *Log filename* field.

- *Connection details* logs inbound and outbound connections and the IP addresses involved.
- *Sender addresses* logs the addresses given as the sender of inbound messages.
- *Receiver addresses* logs the addresses messages are supposed to be sent to.
- *Listserv requests* adds log entries when listserv messages are processed.
- *Errors* adds an entry for serious errors encountered when starting up or delivering a message.
- *Remote SMTP messages* logs the responses of remote SMTP servers to the local server's commands.
- *Delivery messages* gives details about the delivery process in general.
- *All commands* logs all commands given to the SMTP server by remote servers.

Internet Rex's servers

Spool setup

To allow maximum compatibility with different programs, Internet Rex's mail server can store messages in a number of different formats. It can also be configured to use those formats for outbound messages. Deciding what format you want to use and providing the information for it is done in the spool setup menu.

The first entry, *Mailbox directory*, is required no matter what the spool format. This directory is where Rex stores information about what messages a user has. You then select the *Storage spool* format you want.

- “Rex native” is Rex's own storage format: faster than the rest as a result, and more efficient, but not likely to be compatible with any other programs. This format requires that you specify a directory for users' mail to be stored in, the *Rex native spool directory*. This directory must be on a filesystem that supports long filenames; either that, or all the users you enter in the user list must have usernames that fit the standard DOS 8.3 filename notation.
- “UUCP” uses a standard UUCP / UUCICO outbound: .D/.X pairs are created for inbound messages, and .CMD/.XQT/.DAT triples are read periodically to send outbound mail. To use this format you must specify the directory where the .D/.X/.CMD/.XQT/.DAT files will be read and written (the *UUCP spool directory*), the format for the files in that directory (the *UUCP spool type*) and the *UUCP site name* to be used for your machine.
- “KA9Q” stores and reads messages in the KA9Q .wrk/.txt format. This spool type requires nothing but an *Inbound* and an *Outbound directory*. Inbound messages will be stored in .txt/.wrk files in the inbound directory, and the outbound directory will be scanned periodically for .txt/.wrk files to be sent out.
- The “Unix” spool type is meant to allow the mail server to emulate the standard setup for mail on a Unix system. It makes Rex store user messages in a mailbox file in the *Unix mailbox directory*: on most Unix systems this is /var/spool/mail.

Setting up mail users

There are eight different types of mail users available to setup, each providing a different function. To add a user, choose *Setup users* and hit insert. You'll be presented with a list of user types to setup:

- *Normal users* are your regular, everyday mail user: the standard internet email address.
- *Alias* users point to another user you've configured. When mail is sent to the alias, Rex treats it as if it were being sent to the user it points to.
- *Forward to* users are a bit like alias users in that Rex treats mail to the user as if it were to another address. The difference is that the address it's sent to is not on your system.
- *Mailing lists* are a way of sending one message to lots of people, similar to an echomail conference only for the internet.
- *Wildcard relay* users check the destination address of an inbound message, and if it matches the wildcard, the message is sent on to another server.
- *Mail robots* are programs that are run when mail is sent to their address.

Internet Rex's servers

- *Domain store and forward* takes all the mail for a subdomain of yours and stores it in one user's account. These are useful if you want to host mail for someone who can't run an SMTP server of their own.
- *Spam filters* check the source or destination of an address against a wildcard you give. If the wildcard matches, the message is automatically rejected.

Configuring a normal mail user

All a normal mail user really needs is the *Login name* they'll have on your system and the *Password* they'll need to access their mail. The *Login name* represents what's to the left of the @ sign in an email address. So if your domain were somewhere.com and the user's login name were joe, his email address would be joe@somewhere.com.

To provide a few more options for you though, there are additional fields. The *Real name* you enter will be displayed when someone fingers the finger daemon for this user's login name, as will the contents of the user's *Plan file* if you've enabled that part of the finger daemon. You should allow the user some way to modify their plan file: this is a way of letting them have some personalised information displayed when they're fingered.

Use APOP to login is a security measure for your system. You can require a user to use the APOP login method to pick up their mail by setting this to "yes". APOP is a more secure login: it doesn't require that a user's password be sent in the clear over a connection. This way if someone is listening in on the connection, they won't be able to grab the user's password. The only drawback to requiring APOP is that some mail clients don't support it, so enabling it might mean the user can't pick up mail at all.

Finally, if you feel a user's mail might be starting to take up too much space on your system, you can have Rex do a little automatic pruning. Setting the *Message expiry* field to a non-zero number will tell Rex to delete any of this user's messages which are over that number of days old. If that doesn't work for keeping mail trimmed, you can also set a maximum *Mailbox size*. This won't delete any messages, but if the user's mail exceeds the number of kilobytes you give, any new messages they might receive will instead be bounced back to the sender along with a note saying the user has exceeded their mail quota.

Configuring an alias mail user

This is pretty simple. The *Username* is the address for the alias. The *User to point to* is the user mail sent to this alias will actually be delivered to. The *User to point to* field must point to a valid, existing mail user configured in Rex. If it doesn't, mail to the alias will be returned to its sender with an "invalid address" message attached.

Configuring a forward to mail user

Even simpler than an alias mail user. The *Username* is the address for the mail user. The *User to send to* is the address mail will be forwarded to. For example, if the username were joe, your domain were somewhere.com and the user to send to were

Internet Rex's servers

bob@somewhereelse.com, mail sent to joe@somewhere.com would be sent on to bob@somewhereelse.com.

Configuring a mailing list

A mailing list is like an echomail conference for the internet. When someone sends a message to the mailing list's address, that message is sent on to everyone on the list. Because of the range of functions mailing lists serve, there are a lot of options to configure, some in the main screen, some in the *List behaviour* submenu.



The most basic things a mailing list needs are a username (*List name*) for people to send mail to, and some *Users*. Choosing the *Users* field and hitting insert will allow you to add a new user to the list. If you've setup a listserv address (in the *SMTP daemon setup* menu), people can also join the list themselves by sending a join request to the listserv.

Each user requires a valid *Email address* for mail from the list to be sent to. You can also enter the *User's name* if you like, to keep track of which address belongs to whom. Finally, you can also set whether each user *Can post messages* to the list, *Receives messages* from the list and whether messages they post to the list are echoed back to them (*Echo on*). Usually, all of these entries would be set to "yes", but there are some cases where you might want to turn them off. For instance, moderated or informational mailing lists would have *Can post messages* turned off for most of the users; the user who posts messages to an informational list probably wouldn't want *Receives messages* on; and if you've subscribed one mailing list to another, you would turn off echoing of messages so that posts don't end up looping indefinitely.

The last three fields are mostly informational. The *Confirmation date* is of interest to mailing lists with confirmation intervals. It will display the last time a user confirmed they wanted to remain part of the list, or if they're in the middle of confirmation, the number of tries they have left until they're removed from the list. The *Join date* displays the time the

Internet Rex's servers

user joined the mailing list, and the *Last post* is the time this user last posted a message to the list.

Now that you have a list and at least one user, let's examine what happens when a message is sent to the mailing list. When the message is received, the list processor first checks to see if the list is still active (*List active*). If it is still active it next checks to see who sent the message. It can take the address from two possible fields, either the From: field in the message itself or the MAIL FROM field that the SMTP server received when the message was delivered. These can be different addresses: MAIL FROM is harder to fake, but some messages only have the correct address in the From: field. You can determine which field the list processor should use by setting the *Get address from* field.

Once the list processor has determined who sent the message, it checks to see if they're a member of the mailing list. Depending on what sort of list you want to setup, you may not want people to be able post to the list if they aren't members. If that's the case, you should turn off *Outside posting*. You may also want to make your list a *Private list* and turn off *Remote joins*. Private lists aren't displayed when someone requests a list of mailing lists from the listserver. Even if the list isn't displayed, someone can still join your list through the listserver if they know its name: you can prevent this by turning off remote joins for the list. Then the only way someone can be added to the list is if you do it yourself, or the list's remote administrator does.

Assuming the message has come from someone who can post to the list, one of two things now happens to it. If the list has a *Moderator*, the message is sent on to the email address you entered in that field. The moderator then decides what to do with it. If the list doesn't have a moderator, or the message is from the moderator, it gets posted to the list. You can have additional text inserted before or after the body of a message by specifying a file containing the text you want to add in either the *Header file* or *Footer file* field. If you've given a *Digest path*, the message is also saved in a file in that directory. Mailing list digests are an excellent way of keeping a record of the posts to your lists.

If all goes well, the message is sent on to the members of the list to read. Sometimes errors occur though, for instance if someone joins the list then changes email addresses. When someone on the list becomes unreachable, an error message is sent to the address you specify in the *Errors to* field.

An alternate way for people to read your mailing list is to receive a daily digest of the posts. You can enable this by setting the *Digest days* for the mailing list: this sets what days digests will be sent out. Then if a user of the list has *Digest* set to "yes", instead of receiving each post as it comes through the server, they'll receive a digest of all the posts over the last while on the days when digests are sent out.

Sending and receiving messages isn't necessarily all that's involved in setting up a mailing list though. If you've setup the listserver, each list should be configured to interact properly with it. When a user sends a message to the listserver asking to be added to your mailing list, the server first checks the list's *Remote joins* setting. If remote joins are allowed, the server sends a quick message to the address in the *Notifies to* field to inform them that someone has joined the list. The user is then added to the list and their *Can post messages* and *Receives messages* fields are set the same as the *Posting on* and *Receive on* fields for the list. Finally, you can configure your mailing list to send a welcome message back to users

Internet Rex's servers

when they join through the listserver. Simply specify the name of a file in the *Welcome message* field, and that text will be sent back when users join.

If you find yourself away from the computer a lot, or you want to allow someone else the chance to oversee a mailing list, you can configure a remote administrator for your mailing list. Entering the *Remote administration* menu, you can either enable or disable remote administration for each list. Enabling it will require you to enter the email address (*Administration address*) of the remote administrator and the password (*Administration password*) they must give to have their commands accepted. With those in place, the remote administrator can now add or remove users from the mailing list using the administrator listserver commands. (See the file `listserv.txt` included with Rex for more details on listserver commands.)

The final two fields for mailing lists allow you to require that users on the list confirm they want to stay on the list every once in a while. Set the number of days between confirmation requests in the *Confirm interval* field, or return it to zero to disable confirmation requests. When the time interval is up, the listserver will send out a message asking the user to reply to confirm they want to stay on the list. The message will be sent 3 times over the next 5 days. When the 5 days are up, the user will either be removed from the list completely or their *Can post messages* and *Receives messages* fields will be set to “no”, depending on whether you set *Action on expiry* to “remove” or “disable”.

Configuring a wildcard relay

A wildcard relay can be used to override the usual mail routing scheme for a particular set of email addresses. To do this you specify a *Relay wildcard*: a wildcard to use to identify the set of addresses you want to route elsewhere. (e.g. `*@aol.com` would match any email destined to someone at aol.com.) You then give the IP address or machine name of the *Destination host* you want to receive the re-routed mail. This host must be running an SMTP server, though it need not be on the standard port. (If you want to connect on a port other than 25, add `:<port>` to the end of the destination host. E.g. to connect to mail.aol.com on port 1025, you'd set the destination host to “mail.aol.com:1025”.)

Rex will still apply relaying rules to inbound mail that it receives, even if it matches a wildcard relay entry. So if you had excluded particular hosts from relaying through your SMTP server, even if mail from that host matched a wildcard relay entry, it would be refused. You can override this behaviour by setting the *Forced relay* field to “yes”. In this case, mail will always be relayed to users matching the relay wildcard, regardless of its source.

Configuring a mail robot

A mail robot user runs a program every time mail for its address is received. You could use a mail robot for generating a particular set of information every time someone sends mail to an address, processing orders or commands through email, and so on.

The robot first needs a user name to receive mail at: the *Robot name*. Then it needs a program to run to process the mail: enter this in the *Command line* field. In the command line, you can enter the macros `%msg%` and `%dlv%`: these will be replaced with filenames. The

Internet Rex's servers

filename from %msg% will contain the entire text of the message sent to the robot, including all the headers. The filename from %dlv% will contain delivery information for the message: the first line will be the email address of the sender, and any lines after that will be email addresses the message was to be delivered to. (The robot's email address will always be one of those the message was delivered to.)

The robot can insert new messages to be delivered by writing to Rex's mail spool. If Rex is setup to use its native spool, the robot should create two files in that directory: the first should have a name matching *r*.dlv* and should follow the format of the delivery file it received. That is, it should be a plain text file where the first line is the email address of the sender, and the remaining line or lines contain the address(es) of the recipient of the email. The second file should be named *r*.msg* (where the *r** matches that of the delivery file) and should contain the text of the email message. Newlines for the message must be given in CR/LF format or the message may be garbled when sent.

Configuring a domain store and forward user

Domain store and forward users receive mail not for just one address but for all addresses in a domain that you specify. Naturally, you must have control of that domain yourself. Configuration of a domain store and forward used is very much like that of a regular user (see above), with the exception of two additional fields which must be setup.

The *Domain* is the internet domain that this user will receive mail for. For instance, if you give the domain as "joe.somewhere.com", any mail the SMTP server receives for a user@joe.somewhere.com will be put in this user's account. Whether this user would also be allowed to receive mail for, say, user@sub.joe.somewhere.com (a subdomain of joe.somewhere.com) is controlled by the *Subdomains allowed* field. If subdomains are allowed, the mail will be put in the user's account – if not, the SMTP server will reject the mail outright as an invalid address.

Configuring a spam filter

Email spam is, unfortunately, everywhere. You can do a small part to reduce the spam received on your server using spam filters. These mail users don't receive mail, they block it.

To use one, you give a wildcard to use to match against email addresses the SMTP server would receive. (For instance, *@aol.com would apply to users of America Online.) You then decide what field the filter should be applied to in the *Field to apply to* field. "RCPT TO:" will compare the wildcard you gave to the recipient of the email and will reject the message if it matches. This is useful if you have an address that is being flooded with spam and you want to cut off email for it for a while. "MAIL FROM:" will compare the wildcard to the email address of the person sending the message and reject any message they might send if it matches. Use these sorts of filters to prevent known spammer addresses from sending spam to your users. Usually the addresses spammers give don't exist to begin with, so MAIL FROM filters are great for blocking spam. (eg. "joe@hotmail.com", "*@*.newyou.com", "*@makemoney.com", and so on.)

Internet Rex's servers

Setting up the finger daemon

The finger daemon is an internet standard way for people to find out if they have mail, or find out something about someone using only their email address. The usual way it's used is to `finger joe@somewhere.com`. The finger program then connects to `somewhere.com` on the standard finger port (port 79: enter the port you want to use in *Port to use*) and retrieves information about joe. A response from the Internet Rex finger daemon might look something like this:

```
Username: joe
Real name: Joe User
  Last read mail: January 21, 1999 12:23:10 AM
  User has new mail.
Hi, my name is Joe and this is my .plan file.
It doesn't have much in it now, but if I ever get any free
time, that might change..
```

The information the finger daemon returns is quite configurable: Rex can either display or not display each of the first four lines. The final three lines of the example are the user's `.plan` file, which can be enabled or disabled separately. Line 1 is controlled by the *Show user name* field, line 2 by the *Show real name* field, line 3 by the *Last read mail* field, and line 4 by the *New mail* field. Users' `.plan` files can be displayed or not depending on the setting of the *Plan file* field. Note that in order for a user to have a `.plan` file displayed, you must first enter that filename in the user's *Plan file* field in the user setup menu.

Note that while you can enable or disable all of the fields for the finger daemon, if you consider any of them a security risk you're better off not running the finger daemon at all. It is used for providing information: if you want to hamper it doing that, there's little point to running it.

The ident daemon

Server config → *Configure (Ident daemon)*

The ident daemon provides identification services for your machine. These are used mostly by remote servers (SMTP servers, IRC servers, FTP servers, etc.) to authenticate the user logging in to the service they're providing. This gives the administrators there a measure of security by being able to identify troublesome users on other systems: some servers (mainly IRC servers, but some SMTP servers too) will not accept connections from systems without authentication from an ident daemon.

The ident daemon simply returns a user name. The user name it returns is configurable: enter it in the *Ident username* field. Anytime a server requests authentication, that is the user name that will be returned. For single user systems like OS/2, Windows 95 or Windows 98, this should work fine with what's setup. WinNT users may find it works well too. Linux users will probably want to use the OS native ident server instead of Rex's server, as it will more accurately reflect which user is actually using the service being requested.

Internet Rex's servers

The BinkP daemon

Server config → *Configure (BinkP daemon)*

Until version 2, Internet Rex was limited to being a BinkP client: it could connect to BinkP servers, but those servers couldn't connect to it. The BinkP daemon changes that, providing a full-featured BinkP daemon completely compatible with all other major BinkP programs.

The BinkP daemon configuration includes a few fields familiar to those who've setup BinkP connections in the node manager: others are specific to the server. The most important of these is the *Port to use*. This tells Rex what port it should listen to for incoming BinkP connections: the usual port is 24554, but this is by no means required. Many servers operate on different ports, and most clients offer ways of connecting to servers on non-standard ports. If you want to use a port other than 24554, you must nevertheless let your downlinks know the new port to connect on.

The *Security* field lets you decide what level of security Rex should provide on incoming BinkP connections. The normal level is "Standard (none)". This allows any BinkP connect to pickup mail for the addresses it presents. This can be a bit of a security problem if you have people polling with addresses they don't own. To overcome this, there are two further settings: "Deny unlisted connections" and "Deny unpassworded connections". "Deny unlisted connections" will reject any incoming call where the only addresses presented aren't in the node manager or in the nodelist with a BinkP flag. "Deny unpassworded connections" will deny every incoming connection unless it presents a valid session level password. This is the most secure, but can cause problems since nodes must contact you and establish a session level password before ever being able to connect with your system.

Like a standard BinkP connection, you must provide default ways to configure the connections Rex will be receiving. The *Default domain* is used to make all your system addresses into 5D addresses, since BinkP requires 5D addresses when establishing a session. If you've specified domains for your system addresses, this field will be ignored: otherwise, it will be added to the end of each address your system owns. The *Packet size* governs the size of packets that will be used to transmit information across the BinkP link. A smaller packet size is less efficient, but will work better on poor connections: a large one will send files faster on good links, slower on poor ones. You can enter any number between 1024 and 32768 bytes. Finally, should a connection go particularly badly, or should something odd happen, it will be terminated if no activity occurs for a certain amount of time. You can configure that time limit in the *Timeout* field.

The BinkP daemon can keep a separate history log of the connections it receives. Enter the name of the file to log sessions to in the *History file* field, or leave it blank if you don't want to generate a log. The file format will be the same as the log type you entered for Rex's history logs (in *Files, paths and programs* → *Internet Rex files and paths* → *History file format*). The task that will be used in .HIS format files is set in the *Task to use* field and the number of lines of log to keep is set in the *Lines to keep* field. History log files will be trimmed back to this many lines every night at midnight.

Tips and tricks

- BinkD can be configured to send mail out to particular nodes whenever mail is available. Rex can be made to act the same way by setting *Rex's behaviour* → *Crash/IMM mail* to “Send immediately” and setting *Crash all mail* for the node you want mail sent to to “yes”.
- If you want to setup a session level password for a node who isn't in the node manager, you can add them in as a regular node by setting the session level password (in *Connection details* → *Session password*) and leaving the *Inbound mail* and *Outbound mail* options at “Don't toss inbound/outbound mail”.

Troubleshooting

“Anything that can go wrong, will.”

- *Murphy's first law*

In an ideal world, every program, once installed, would run perfectly and there would never be any problems. The real world works a bit differently.

Rex will usually give you a pretty good idea of what problems it's having when things go wrong. How to fix them may not be immediately obvious. To help with that situation, this section will present a list of common problems and how to fix them, as well as a list of errors Rex might give you, what they mean and how to fix them.

Common problems

I tell Rex to send files, but nothing gets sent, even though I know there's stuff there to be sent.

Check the setup for your frontend mailer in *Files, paths and programs* → *Frontend mailer setup*. You should also check the setup of the nodes you expect Rex to find mail for: make sure the node hasn't been set inactive, that you have the right netmail address and that any routing commands you've setup are correct.

If you're connecting to the net over a dialup connection and you're running under Windows, check that the connection you've pointed Rex to is actually there, and that Rex can access the modem. If *Autodial* is set to “no”, remember that you'll have to establish the connection yourself before Rex will use it.

Rex keeps downloading email from person A and saying it's from person B.

or

Rex is downloading email from person A and saying it's a stray Internet Rex / Fido2Int / Allfix / TransX message.

Misidentification of mail is almost always the result of incorrectly setup matching rules. It's very important that the matching rules you setup for an email node match that node's mail and only that node's mail. The first problem indicates the matching rules for person B may be too broad (matching mail that it isn't supposed to), and/or the rules for person A are too narrow or incorrect (they aren't matching against person A's mail). Similarly for the second problem, the matching rules for person A are too narrow or incorrect.

You should setup matching rules first so that they match the person's email address in the From field. That is, for someone whose email address is “joe@somewhere.com”, the first matching rule should be

From contains joe@somewhere.com

If Joe is likely to send you email from that address which you don't want Rex to decode, have him pick a subject he'll use to send mail which Rex should be decoding. (A good choice might be “Mail for Rex” or “SomeNet mail”.) Then add an extra rule to the matching rules so you end up with

Troubleshooting

*From contains joe@somewhere.com and
Subject contains Mail for Rex*

For more on exactly how Rex checks matching rules, look in the technical manual.

It's important that Rex be able to correctly link inbound mail to nodes in the node manager: if it can't do that, the inbound mail may be mis-processed, or there may be errors generated which normally wouldn't apply to the message.

I'm using the Windows version of Internet Rex with dialup networking, and when I go to start Rex, all I get is Windows telling me another application is using the modem.

Windows (especially Win95) isn't that bright when it comes to dealing with multiple programs trying to use the modem. Certain DOS programs (such as FOSSIL drivers or some terminal programs) will result in Windows thinking the modem is in use long after they've been shut down, and so long as Windows thinks the modem is in use, it won't establish a dialup networking session.

There is a fix for this problem. You'll need to open the file `system.ini` in your Windows system directory with your favourite text editor, and find the `[386Enh]` section. Add the line

```
Com<n>AutoAssign=0
```

to the file, save and reboot. Replace the `<n>` with the COM port your modem is on. So for instance, if your modem were on COM2, you'd add in

```
Com2AutoAssign=0
```

This tells Windows to allow multiple programs access to the modem at once. You can also set this value to anything between 1 and 999 to have Windows wait a certain period of time before transferring control from one program to the other.

I'm using the DOS 16-bit version of Rex and using a BETWEEN.BAT or automatic packing/unpacking of mail for a node. I frequently get memory manager errors. (QEMM exceptions, etc.)

The EMS/disk swapping routines included with Rex seem to react badly with some memory managers. If you're having these problems, try disabling EMS/disk swapping (in the *Rex's behaviour* → *OS dependent options* → *DOS* menu). The programs you run (BETWEEN.BAT and archivers) will have about 200K less conventional memory to run with, but the memory manager errors will cease. If this causes problems, try running the 32-bit version of Internet Rex for DOS. It uses a different system for memory management, one that might be more compatible with your memory manager's.

Errors from Internet Rex

This is a list of some of the most common errors you're likely to encounter with Internet Rex, what they mean, and how to fix them, if applicable.

Troubleshooting

Fatal errors (!)

Rex died from internal error 5 (non-Linux versions)

or

Rex died from internal error 11 (Linux version)

Usually this indicates a bug in Internet Rex. Send in a bug report, including a copy of your log file and your *.REX files.

Rex killed by user with signal 4

You pressed Ctrl+Break or sent a kill signal to Internet Rex to stop it running. This is nothing to worry about, but Rex gets a little concerned when it happens.

Rex died from internal error # (# is anything other than 1 to 9)

This often happens under DOS when Ctrl+Break is pressed, as signal handling works a little different there. If you didn't press Ctrl+Break to stop Rex, this is a sign of a bug in Rex. Send in a bug report, including a copy of the log file and your *.REX files.

Other errors (?)

Incoming file did not have an associated manifest record

The message Rex received was from a node that had a reliable connection setup (or at least had *Send acknowledgements* set to "yes" or "batch") but the inbound file wasn't in the manifest, or there was no manifest to begin with. Usually this happens when you've setup a reliable connection with another node, but the other node hasn't enabled a reliable connection at his end. No files will be lost, but you should check with the other node to get the connection setup correctly. If the other node has set up auto-resends, not fixing the connection may result in you receiving duplicate files and/or messages.

Incoming file failed CRC check

This may happen on occasion if a message gets altered between someone sending it to you and you receiving it. If it can, Rex will request a resend of the file so the data arrives correctly. If you get these repeatedly from the same node, it may be a sign that the other end has setup encryption and you haven't, or that your encryption passwords don't match.

Incoming file failed authentication check

This may also be the result of a message being altered between someone sending it and you receiving it. If you receive these repeatedly from the same node, it may also be a sign that your session passwords don't match, or if you haven't got a session password setup for that node, that the other node has set one up without telling you about it.

FTP script failed at line # with error #

There was a problem with the FTP script for this node. The number after "error" indicates what the problem was.

Troubleshooting

- 1 The FTP site stopped responding to commands, or data was not received when it was expected. This will happen if the connection to the FTP site is lost, either at your end or somewhere else along the line, or if the FTP site goes down while you're connected.
- 2 The FTP site disconnected while you were online. This can be the result of the site deciding something you were doing was bad, or just the connection going dead. Try running your script manually to make sure that it's not doing anything the FTP site is opposed to.
- 3 A file transfer was attempted, but the connection to transfer the file over was dropped or never established. This may be the result of a poor connection between you and the FTP site.
- 4 Not enough space. You tried to upload something to the FTP site, but it ran out of space.
- 5 Permission denied. There are any number of permissions that you might not have had, including no access to uploading, downloading, deletion or reading a directory. Whatever command your script was running on that particular line was doing something the FTP site objected to: check your directories and permissions on the FTP site.
- 6 Remote error. The FTP site had a problem, unspecified. Probably won't happen again. Probably....
- 7 Bad filename. The file or directory you tried to create or upload had an invalid character in it, or was just too long. Some FTP sites run on FAT filesystems, which don't allow anything except DOS's usual 8.3 filenames.
- 8 Command unsupported. Usually happens with commands sent through QUOTE. Basically, the FTP site didn't support the command you sent.
- 256 Unknown response. The FTP site responded in a way Rex didn't understand. This may happen repeatedly, in which case you should send a bug report with a copy of the log file with debug logging turned on.

Remote didn't understand SEAT keyword....

You're using a SEAT connection with a node whose implementation of the SEAT standard is a little older or a little less compliant than Internet Rex's. Rex will generally try to adjust its setup for that node so it doesn't use that particular keyword again (this applies to the "Auth", "Crypt" and "Freq" keywords). Some misunderstood keywords may be a result of a SEAT mailer that can't talk to Rex in the mode you have it setup in right now: for instance, Rex can send and receive SEAT messages using the Base64 encoding, but not all SEAT mailers are required to do this. Make sure that you've setup a connection with the remote that both mailers are capable of supporting.

Frequently asked questions

Can I use Rex to give my BBS users access to Internet email?

Yes, you can. To do this, you need to have the authority to assign email addresses to an internet domain. If your computer has an assigned name on the internet (eg. xanadu.v-wave.com), you already have control of an internet domain: all you need to do is be able to run a mail server on your machine. Alternatively, there are name redirection services (eg. tzo.com, dyndns.com, dyn.cx) which can give you an internet domain and mail control.

Once you have your domain, all you need to do is setup the appropriate gateway in the *Gateway setup* menu and start import/exporting your mail. For more details on this, read the [BBS ↔ Internet gateways](#) section of this manual.

Can I use my America Online (AOL) account with Internet Rex?

At last check, AOL didn't provide you access to their mail servers directly from your machine: you had to be using their software to send or receive email. Because of that, Rex can't send mail through your AOL account.

If AOL change their setup so that you can access your AOL mail through a POP3 or SMTP mail server, that changes the answer to this question to yes. You may want to ask one of the administrators at AOL if access to a POP3/SMTP server is available.

An alternative to this is to use a drop mail service like Hotmail to deal with your BBS mail instead of AOL. These services usually allow POP3/SMTP connects and Rex can access them using your AOL connection.

Do I have to register Rex once for each operating system I run it under?

Nope. If you register Internet Rex, the registration code you receive will apply to all versions of Internet Rex. For instance, registering the DOS version means you'll get a registration code that works with the DOS, OS/2, Windows or Linux versions of Internet Rex.

The only limitation on registration codes is that they are linked to the *Sysop's name* field in General information. You and only you can use your registration code. If you want to transfer your registration to someone else, please contact the person you registered Internet Rex with.

I run a DOS BBS system under another OS (like Windows, OS/2 or Linux). Can I use the native version of Rex for that OS, or do I have to use the DOS version? How do I start the native version of Rex from my DOS BBS?

Most operating systems with DOS emulation support provide a way of starting native programs from DOS, so you should be able to use the native version of Internet Rex with your DOS BBS.

For Windows users, this is done with the `start` command. To launch a native Windows application from a DOS box and wait until it's done, use the command:

```
start /w (program and parameters)
```

Note that this will only work if you are running from a DOS box under Windows: if you've told Windows to reboot to MS-DOS mode, you're basically now running just DOS and you can't run any Windows native programs. Also note that there can be issues with multiple

Frequently asked questions

programs using the modem at once if you are using dialup networking: see the [Troubleshooting](#) section for more details on this.

OS/2 users should look for the utilities `hstart` or `os2exec`. These let you launch OS/2 native applications from a DOS window.

Another possibility is running Rex in daemon mode (see [Daemon mode](#)). This skips the step of having to launch Rex from your mailer: it just sits running in the background and launches itself when you tell it to.

I accidentally downloaded some mail that Internet Rex was supposed to process. How do I get Rex to deal with it now?

This can be an easy or a difficult thing to deal with, depending on what program you used to download the mail. If you downloaded the mail using a mail program Rex supports (Eudora, PMMail, Postroad mailer, MR/2 Ice or Nettamer), it's a relatively easy problem to fix.

To have Rex process the mail, go to the *Email setup* menu and choose the email address you used to download your mail. Remember the settings you have for the configuration in *Mail spool type*. Then change the mail spool type to the program you used to download the mail, and set the directories appropriately. (You may want to take a look at the [Email setup](#) section for help with this.) Run Rex with the `-fetch` command, and it will process the mail you downloaded. Then change the email address back to the way it was.

If you didn't use an email program Rex knows about, things are a little more complicated. You still have to go to the *Email setup* menu, choose the email address you downloaded the mail with and write down the settings for *Mail spool type*. This time, change the mail spool to "KA9Q style SMTP spool", and point the inbound and outbound directories to a couple of temporary directories on your system. Remember what you entered for the inbound directory.

Now, for each of the messages you want Rex to process, save the text of the message, including the headers, to a file called `#####.TXT` in the inbound directory you gave above. (Replace `#####` with a number of your choosing.) When all the messages are saved, use a text editor to create a file called `#####.WRK` for each of the `.TXT` files you made (the `#####`s have to be the same). In this file, put the lines:

`From: sender's address`

`To: your email address`

Replace *sender's address* with the email address of the person who sent that piece of mail, and *your email address* with your actual email address. When you've created a `.TXT/.WRK` pair for each message, run Rex with the `-fetch` command and it will process the messages. Then you can remove the temporary directories and change the mail spool type for your email address back to the way it was.

Copyright notice

Internet Rex is copyright (1997-2000) of Khan Software and Charles Cruden. Permission is given to use Internet Rex for a period of 30 days without restriction. After that time you must either register the program or remove it from your system.

To register Internet Rex, look for REGISTER.ZIP in your original Internet Rex archive and use the file appropriate for your country.

Internet Rex is provided 'as is', with no warranty expressed or implied. The only thing Rex is guaranteed to do is occupy disk space. No technical support is guaranteed, though I will do my best to provide help when possible. See the [Support](#) section for information on contacting the author.

Under no circumstances will Charles Cruden or Khan Software be held responsible for damage caused by running Internet Rex or its associated problems. Any loss, financial or otherwise that may have been incurred while running, or as a result of not being able to run, Internet Rex is the sole responsibility of the person using the software.

Users who chose to register Internet Rex will receive a serial number and registration key which are unique to their system. Under no circumstances are these to be distributed to anyone other than the user they were originally intended for. Use of these keys on a system other than the original registrant's is expressly prohibited.

Failure to comply with these limitations may result in the registration being revoked.

Internet Rex and its associated programs may not be hacked, reverse engineered, run-time modified, etc. Distribution of patches, key generating or bypassing routines or any similar products is expressly prohibited and will result in the revoking of registration keys for people caught doing so. It may also result in legal action.

Users who have registered Internet Rex but later decide they no longer wish to comply with these restrictions may withdraw from them. In so doing, they forfeit their registration for all future versions, and their registration fees.

Other programs mentioned in this help guide are copyright of their respective owners.

Registration

The distributed version of Internet Rex is the shareware version. You are free to use this version for 30 days as a trial run of the program. If after that period, you continue to use Internet Rex, it has probably met with your approval and you are requested to register it.

Internet Rex has two registration choices for a total of four registration options. You can choose to register either 50 or 1000 nodes. Only the 1000 node version has the ability to send to nodes not listed in the node manager. The 50 and 1000 node versions also have two choices: regular and server. Regular versions can run the FTP, POP3, SMTP and finger daemons, but are limited to two FTP and two mail users. The server version can configure as many users as nodes registered (50 users for the 50 node version, 1000 for the 1000 node version).

Registration of Internet Rex can be done at any of the registration sites listed in the forms in the archive `register.zip`, distributed along with the rest of Internet Rex.

If you want to purchase large numbers of registrations for Internet Rex at once (5 or more), it's possible a discount can be arranged. Similarly, if there is no registration site for your country, it's possible an arrangement can be worked out for payment in your currency. Either way, please contact the author to work out details. (See the [Support](#) section of this manual.)

Index

A

- AdeptX
 - Configuring Rex with AdeptX · 23
- Allfix
 - Settings for an Allfix node · 40

B

- BBS
 - Downloading Rex from a · 9
- Beemail
 - Configuring Rex with Beemail · 26
- BETWEEN.BAT
 - Command line option · 50
 - Configuring Rex to use one · 81
- Binkley
 - Configuring Rex with Binkley · See Frontend mailers, Binkley and compatible
 - Setting up multiple outbounds · 27
- BinkP
 - Connecting to a BinkP site · 42

C

- Colour schemes · 74
- Command line
 - Configuration program's command line · 92
 - Rex's command line · 48
 - Syntax · See Syntax of commands
- Connecting to the net · 33
 - DOS · 34
 - Linux · 34
 - OS/2 · 36
 - Windows · 35
- Conventions · 5

D

- D'Bridge
 - Configuring Rex with D'Bridge · See Frontend mailers, D'Bridge
- DOS
 - OS dependent options · 18
 - OS specific command line options · 52
- Downloading Rex · 9
- Drive mapping · 76
 - Mapping drives to directories · 77
 - Mapping drives to drives · 76
- Duplicates

Detecting duplicate files · 84

E

- Email file requests
 - Downloading Rex via · 10
 - Rex's email file request processor · 80
- Encryption · 89
- Error messages · 114
- Eudora · 32

F

- Features of Internet Rex · 1
- Fidonet2Internet mailer
 - Settings for a Fido2Int node · 70
- File request
 - Downloading Rex via · 9
- File requests
 - Rex's file request processor · 77
- FrontDoor
 - Configuring Rex with FrontDoor 2.25+ · 25
 - FrontDoor up to 2.20 · See Frontend mailers
- Frontend mailers · 22
 - AdeptX · 23
 - Beemail · 26
 - Binkley and compatible · 23
 - D'Bridge · 23
 - FrontDoor 2.25+ · 25
 - FrontDoor/APX · 26
 - KBBS · 24
 - PCBoard · 24
 - Platinum Xpress · 24
 - QFront · 25
 - T-Mail · 23
 - Unidentified (*.PKT) · 26
 - ViaMail · 23
- FTP
 - Downloading Rex via · 9
 - Passive mode FTP · 27
- FTP server · 44, 45, 94

G

- Gateway, Fido/Internet · 30
 - Echomail to domain gates · 65
 - Echomail to email gates · 64
 - Net gates · 66
 - Netmail to domain gates · 63
 - Netmail to email gates · 63
 - Using Rex as a gateway · 56
- GIGO
 - Settings for a GIGO node · 40

Index

H

- Help
 - Command line help · 51
 - Online help · 14

I

- Installation · 11

K

- KA9Q · 31
- KBBS
 - Configuring Rex with KBBS · See Frontend mailers, KBBS

L

- Linux
 - FTP server requirements · 98
 - OS dependent options · 18
- Locks
 - Directory locking on FTP sites · 44
- Logging
 - Configuring Rex's log · 84
 - Trimming the log file · 50

M

- Mailing lists
 - Gateway · 66
 - Gateway listserver · 67
 - SMTP server · 106
- Main menu
 - Address manager · 15, 26
 - Connecting to the net · See Connecting to the net
 - Email setup · 28
 - Files, paths and programs · 20
 - FTP setup · 27
 - General information · 16
 - Logging options · 84
 - Queue editor · 85
 - Rex's behaviour · 17
- Manifests
 - "No associated manifest record" · 115
 - Configuring Rex to use them · 87
- Matching rules · 41
- MR/2 Ice · 32

N

- Nettamer · 32, 33

O

- OS/2
 - OS dependent options · 18

P

- PCBoard
 - Configuring Rex with PCBoard · See Frontend mailers, PCBoard
- Platinum Xpress
 - Configuring Rex with Platinum Xpress · See Frontend mailers, Platinum Xpress
- PMMail · 31
- POP3 server · 29, 100
- Postroad mailer · 31
- PXFids
 - Settings for a PXFids node · 40

Q

- QFront
 - Configuring Rex with QFront · 25
- Quick setup guide · 15

R

- Routing · 38, 54, 72

S

- Secure connections · 90
- Setting up a new node · 38
- SMTP server · 29, 102
- Soup / Yarn · 31
- Statistics
 - Command line options for · 92
 - Viewing them in the configuration program · 90
- Syntax of commands · 6

T

- Technical support · 8
- T-Mail
 - Configuring Rex with T-Mail · See Frontend mailers, T-Mail
- TransNet
 - Settings for a TransNet node · 40
- TransX
 - Setting Rex's TransX mode · 19
 - Settings for a TransX node · 40

Index

U

Upgrading · 12
UUCP · 30

V

ViaMail
Configuring Rex with ViaMail · See Frontend
mailers, ViaMail

W

Watergate
Settings for a Watergate node · 40
Wildcards
Rex's use of · 6
Windows
OS dependent options · 18
OS specific command line options · 52
WWW
Downloading Rex via · 9